Title
In Search of a “Fair Explanation”: Helping Young People to Consider the Possibilities, Limitations, and Risks of Computer- and Data-Mediated Systems

Author
Van Wart, Sarah Jane

Publication Date
2019

Peer reviewed|Thesis/dissertation

In Search of a “Fair Explanation”:
Helping Young People to Consider the Possibilities,
Limitations, and Risks of Computer- and Data-Mediated Systems

By

Sarah Van Wart

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Information Management and Systems

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Coye Cheshire, Chair
Professor Paul Duguid
Professor Kris Gutiérrez

Fall 2019

ABSTRACT

Significant resources have been directed towards K-12 computing and data education over the past ten years, as part of what has come to be known as the *CSforAll* initiative. This initiative has focused on raising awareness of computing education among parents and students, developing situated learning progressions that resonate with many different interests and pursuits, training teachers, and addressing issues of underrepresentation in computing among females and racial minorities. In this dissertation, I argue that as the *CSforAll* initiative continues to expand, it is important for the education community to also reflect on the forms of knowledge that are believed to be essential, and the presumed benefits of computing and data education. Specifically, how might the goal of producing citizens with robust computing and data literacies change what is considered to be fundamental to a computing education; as well as the kinds of contexts in which computing and data science are situated?

I use the term *sociotechnical literacy* to name this vision for computing education, which I define as a broad set of social and technical practices, strategies, ideas, and dispositions that can help people to reason about the computer-mediated systems that shape their everyday lives. As the term suggests, I argue that it is important for learners to engage with technical ideas as well as their social applications and implications. To examine what this might mean for teaching and learning, I describe two design experiments that I conducted with young people (ages 14 – 22). Each approach aimed to make the *applications* of computing primary (rather than treating applications as the backdrop from which the abstractions of computation are motivated), so that learners could examine some of the specific ways in which data and computing might be directed to particular goals, subject to real possibilities and constraints, and in relation to alternative forms of participation.

I examine the possibilities and limitations of each approach. I also analyze some of the assumptions that framed the design experiments – which were naïve, but also reflective of a broader ethos that pervades *CSforAll*. I reflect on what these studies collectively reveal about the possibilities, limitations, and risks of data and computing, as situated in the lives of young people; as well as what this might mean for helping young people develop a robust sociotechnical literacy. There are very real limits to what can be accomplished with computing and data alone. There are also significant benefits *and risks* associated with the many sociotechnical systems that shape our lives. As such, I argue that rather than positioning computing education as a remedy to various social ills, we instead offer young people a fair explanation of what computing is and is not capable of, grounded within specific contexts involving real people. I conclude with what this fair explanation might include, and how it might be fostered.

CONTENTS

ACKNOWLEDGEMENTS

This dissertation, and more generally my journey through graduate school, has been a collective accomplishment that would not have happened without the support and inspiration from mentors, colleagues, students, friends, and family. When I applied to the Master's program on a whim, so many years ago, I never imagined that I would be spending the next 10 years of my life as a graduate student at Berkeley. It has been a true privilege, and an experience that I am so very grateful to have had.

To Tapan Parikh, my advisor: what a journey it has been. Knowing you has absolutely changed the course of my life, and has opened up so many opportunities that I never could have imagined. I have so many fond memories of doing field work with you and the rest of the research group – from co-teaching 10th grade Sociology, to running a summer youth program, to traveling to NorCal in the middle of a wildfire to do some mapping. I appreciate your optimism and creativity, your willingness to embrace new ideas and perspectives, and your support of my intellectual development. Thank you for your patience, for believing in me, and for encouraging me throughout my time in graduate school.

To my committee – Coye Cheshire, Paul Duguid, and Kris Gutiérrez – you have each been such a reliable source of insight and support for me.

Coye, I am very grateful for your optimism, humor, and support with writing, argumentation, and research methods over the years. You really pushed me to think about what the actual problem is and why it matters, which was such a simple but powerful way of helping me to articulate and focus my dissertation project. The doctoral research and theory workshop (drtw) that you ran was one of the highlights of my I School experience, and I appreciate the community that you helped to build around academic writing. I think that many of the relationships that I formed with my fellow PhD students were forged in drtw.

Paul, you have always been so patient, generous, and kind to me (and all of the PhD students). You read so many drafts of my work, and offered me so many thought-provoking comments and questions. While I was certainly not a quick study when it came to theory, you always met me where I was, and helped me take the next step forward, whilst engaging with me as if I were your intellectual peer. That is a rare gift, and one that I very much benefitted from.

Kris, thank you for taking me under your wing and allowing me to regularly participate in your seminar group. I appreciate your lens on learning design work, and the way in which you embrace criticality, hope, and possibility all at the same time. Your ideas are very much present in this dissertation. I am also grateful to you for introducing me to your intellectual lineage (e.g., sociocultural theory, critical race theory, CHAT). My view of learning and education would have looked completely different without your guidance and insight.

I am very grateful to many other faculty members within the UC Berkeley community as well. Kimiko Ryokai, I appreciate your support as I went through my qualifying exam, and your lens on learning and the importance of creativity, expressivity, and art. I am also grateful to Deb McKoy, Kevin Cuff, Maggi Kelly, and George Scharffenberger: each of you has a unique and important approach to community-based research, which has inspired me and given my own work direction.

I am grateful to the many PhD students who have helped me work through ideas or with whom I have collaborated on projects. To my two close research collaborators, Kathryn Lanouette and Sepehr Vakil: I very much enjoyed working with and learning from you. I have also benefitted from the wisdom, feedback, friendship, and good humor of Adam Calo, Ariel Bierbaum, Alyse Schneider, Erin Palmer, Sibyl Diver, Anna Keune, Ryan Pettiway, Christyna Serrano, Joanne Tien, Jenni Higgs, Kuang Cheng, Kurtis Heimerl.

Within the I School, my intellectual home, my fellow PhD students and friends were a tremendous source of support and intellectual inspiration for me. I am so grateful to Meena Natarajan, Neha Kumar, Anne Jonas, Andy Brooks, Nick Doty, Elaine Sedenberg, Richmond Wong, Ishita Ghosh, Ashwin Matthews, Max Curran, Shazeda Ahmed, Jen King, Vibhore Vardhan, Seongtaek Lim, Nick Merrill, Bob Bell, Laura Devendorf, Sebastian Benthall, Elisa Oreglia, Elizabeth Goodman, Nick Chong, Daniel Griffin, and Elizabeth Rezor. I also owe a tremendous debt to Catherine Cronquist Browning and Inessa Gelfenboym, who offered me substantive logistical and emotional support throughout the dissertation writing process.

From the *OSMO* community, I am grateful for the support of my fellow mentors, particularly Krista Cortez, Blanca Gamez-Djokic, Dawn Williams, Alea Holman, Aaron Hobson, Kene Akametalu, and Jeremiah Sims.

From *Code 510*, one of the highlights of my graduate school experience, I am grateful to Noura Howell, Ricky Holtz, Lucio Lopez, and especially Ellen Van Wyk for co-creating a beautiful space of teaching and learning with young people.

From the *Local Ground* development team, I am grateful to Joyce Tsai, Christy McCain, Zachary MacHardy, Karin Goh, Ellen Van Wyk (again), Cynthia Mah, Riley Flynn, John McGrath, Lynn Deregowski, and Megan Nelson. Together, we built a pretty great digital mapping tool.

From Berkeley City College, I am grateful for all that I learned about teaching and learning in a truly diverse and supportive space. Thank you to my colleagues Justin Hoffman, Mary Clarke-Miller, and Rachel Mercy Simpson; to my co-instructor and fellow I-Schooler Ricky Holtz (again); and to the many students with whom I worked – particularly Lynn Deregowski, Rachel Smith, Megan Nelson, Alexandra Lukinicheva, Noi Smith, Liliana Velazquez, Rosemary Jason, Karin Hauck, Mary Luong, Ehidin Borbon, and Colin Dodsworth.

I am also grateful for the sage wisdom and guidance of Laura Ortiz-Guillen and Natalie Dubois, as I navigated the PhD process.

From Northwestern, I definitely want to thank the Department of Computer Science, for their support in helping me to finish this dissertation – particularly Sara Sood, Steve Tarzia, Larry Birnbaum, Samir Kuller, Willie Wilson, and Chris Riesbeck.

To the many young people with whom I had the privilege to work with: while I can't name you, you know who you are. You were some of my greatest teachers, and some of you were like family to me. I think of you often, and I miss you all. I hope that many wonderful things come your way.

To my family – Mom, Laura, Andy, Dad, Vera, and Karlo – I have felt very supported by each and every one of you during graduate school. Thank you for your guidance and love. Mom, because this dissertation has pushed me to reflect on my own educational trajectory, I have also had the joy of revisiting the many ways in which you supported me from childhood to adulthood. I often heard your voice as I was in the field and when writing. While my advisor, committee, and colleagues provided tremendous intellectual guidance over the course of my dissertation research, you have a strong presence in this dissertation as well.

And finally, to Christy, my wonderful wife: thank you for supporting me through the PhD process in its entirety. You helped me write software for my design experiments. You did fieldwork with me. You read my drafts. You came to all of the many high school final presentations and showcases. And you've helped me to keep it all in perspective.

This dissertation is dedicated to the late Malik Joshua Robert Grant:
philosopher, designer, computer programmer, parkour enthusiast, and dear friend.
You are very missed and very loved.

1 INTRODUCTION

Data and computing increasingly shape how we perceive the world, and the opportunities we have to participate in it. However, while we are becoming increasingly adept at developing new computational techniques to gather, process, and disseminate data and information, we are only beginning to understand how these emerging tools, techniques, and infrastructures impact our lives, both individually and collectively. On one hand, new computer-mediated technologies offer many benefits – making it easier to buy and sell goods and services, talk to friends and family members across time and space, disseminate ideas, and generate new insights. These technologies have also led to an increased demand for highly skilled tech workers, as the information economy continues to grow. On the other hand, networked computing also has a darker side – making possible increasingly sophisticated disinformation campaigns, increased surveillance, and the collection of people's personal information (often without their knowledge) at an unprecedented scale and frequency. Moreover, entire classes of jobs, which have historically offered financial stability and security to many people, are gradually being automated away (Sterling, 2019).

Given the possibilities, risks, and uncertainties of these technologies and methods, there is growing interest in educating young people in primary and secondary school about computing (e.g., Wing, 2006), data (e.g., Berman et al., 2018), and the systems they enable (e.g., Rainie & Anderson, 2017). This includes learning the skills and techniques needed to participate in computing- and data-related tasks; as well as understanding the kinds of activities that these techniques enable, amplify, or organize. Reasoning about the ways that data and computing mediate modern life is an important literacy that can help young people, and society as a whole, to more fully realize the potential benefits of these technologies and to mitigate the potential and actual harms that might result from them. How young people (and educators and parents) are supported in examining the value, power, and accuracy of data- and computer-mediated systems will set the stage (at least in part)[1] for whether they will or will not question these systems, or try and make them better.

However, while considerable attention has been directed towards making the abstractions of data and computing more accessible and engaging to young people, less attention has been given to situating some of the applications, social implications, and limits of computing and data. At the current moment, many of the dominant approaches to introducing computing- and data-related ideas in K-12 revolve around creative expression, games, puzzles, and/or simulations (e.g., Jeff Gray, Abelson, Wolber, & Friend, 2012; Maloney, Peppler, Kafai, Resnick, & Rusk, 2008; Sklar, 2007) – applications which are typically quite separate from real

[1] I am not arguing that data and society challenges can be solved by more knowledgeable users alone, however I do think that user education is a productive site of intervention, where learners can be encouraged to question not only the information itself, but also the *systems* they are using.

world practice. While these approaches have been helpful for introducing computing- and data-related fundamentals – which can certainly generalize to other situations – these learning contexts are often so far removed from the real world that they do little to help young people reason about the broader impacts of data and computing. I argue that if computing- and data-related ideas are not grounded in real world practice at least some of the time, then their applications and implications are left to the imagination of the learner – which may or may not reflect the very real possibilities, limits, and contingencies of data and computing. Moreover, if young people do not have opportunities to consider the relative strengths and limitations of these technologies and methods, then they may naïvely accept systems that are not designed with their best interests in mind, or perhaps pursue technical solutions to problems that are best addressed via other means (legal, organizational, etc.).

In this dissertation, I explore what it might mean for educational initiatives to direct more attention to the real world contexts of computing and data. To do this, I draw on empirical evidence from several design experiments, which were intended to help young people to explore how computer- and data-mediated technologies and methods interacted with the broader world around them. Using findings from these studies, I make the case for the importance of fostering a *sociotechnical literacy,* which I define as a broad set of practices, strategies, ideas, and dispositions that enable people to reason about a particular kind of real-world context: the *sociotechnical system*. I define sociotechnical systems as social systems that rely on computer-based information and communication technologies to carry out critical functions. These systems are diverse and extend to many different human activities. On the business/organizational side, these systems can support activities like decision-making (e.g. deciding whether to relocate the factory to Mexico), knowledge production (e.g. analyzing temperature data to understand climate trends), and service delivery (e.g. reporting and triaging service requests). On the consumer side, they mediate a wide variety of everyday practices, including entertainment (e.g. playing networked video games; watching Netflix), shopping (e.g. buying a book on Amazon), and communication (e.g. sharing cat videos with friends and family).

Sociotechnical systems do tremendous work in the world, shaping it in ways that are not entirely knowable nor under the control of any one individual. That said, by examining some of the mechanisms through which these systems operate, I argue that people will be in a better position to question these systems; think through some of the ways in which data and information are (computationally) collected, used, repurposed, and commoditized (and towards what ends); collectively decide whether (and under what conditions) these systems are supporting or eroding important social values; and act accordingly. However to pursue this agenda, it may be necessary to revisit what kinds of computing- and data-related ideas are most important within K-12, as well as the contexts of computing that need more emphasis within educational initiatives.

1.1 What Constitutes a Sociotechnical Literacy?

In order to consider what might constitute a sociotechnical literacy, I draw on literatures within computing, data, and new media literacy education, which each offer a different perspective on which computing- and data-related ideas ought to be considered fundamental. I also pay attention to the contexts in which these ideas tend to be situated; and whether computing and data are treated as a means or an end in itself.[2]

1.1.1 Computational Thinking v. Computational Literacy

One key fault line within computing education is whether there exists a general set of computing "concepts"[3] that everyone should learn within K-12, or whether concepts are domain-specific (and thus inseparable from their applications). For example, is learning a computing concept like *recursion*[4] a generally useful way of thinking about the world, or is it only useful in under a very specific set of circumstances where repetitive calculations are needed? Each perspective has significant implications for teaching and learning about sociotechnical systems. The dominant perspective, *computational thinking*, leans towards the idea of learning general purpose abstractions. This idea, coined by computer scientist Jeanette Wing, asserts that computer science is not only a niche discipline of technical expertise, but also a "universally applicable attitude and skill set everyone, not just computer scientists, would be eager to learn and use" (Wing, 2006, p. 33). There are a number of abstractions at the core of this idea; and much of the debate centers around whether and how these ideas should be translated into broader educational initiatives. Grover & Pea (2013) review and summarize these core concepts, which include:

- Abstractions and pattern generalizations (including models and simulations)
- Systematic processing of information
- Symbol systems and representations
- Algorithmic notions of flow of control
- Structured problem decomposition (modularizing)
- Iterative, recursive, and parallel thinking
- Conditional logic
- Efficiency and performance constraints
- Debugging and systematic error detection

---

[2] An example of computing as an *end* might be learning about variables, loops, and event handlers by making a game. Here, the context (i.e. making a game) is the means through which knowledge of loops, variables, and event handlers is attained. An example of computing and data as a *means* might be learning about environmental science by calculating summary statistics on a dataset. Here, the context (i.e. understanding environmental science) is the end goal, and data science is one means of achieving this goal.

[3] The term "concept" is put in quotes because the idea of a concept is not fully embraced by learning theorists (neither from the cognitive nor situated perspectives).

[4] Recursion is a computer science concept that involves a strategy where functions can recursively call themselves one or more times until a condition is met.

Each of these ideas, derived from computer science, is described as generally useful, portable, transferable, and therefore possible to teach and learn in any number of interchangeable contexts – whether through games, stories, music compositions, sewing, robotics, and so forth. Within this framing, *context* is treated as a backdrop for teaching the abstractions of computation: a way to motivate an engagement and to make ideas concrete. Thus, much of the work of operationalizing computational thinking in K-12 has involved creating a series of engaging contexts – music, apps, games, robots, stories, and so forth – as a way to help young people learn the abstractions of computational thinking.

However, Andrea diSessa, a learning and cognitive scientist, argues that it is not enough to assert that there are general abstractions (which happen to be rooted in computer science) without understanding the significant integration work that is required to productively marshal these taken-for-granted computational concepts in a particular domain (diSessa, 2018). In other words, just because someone learns how to make a game using certain computing concepts, that does not mean that these "concepts" can be seamlessly lifted up and moved into a new context. Moreover, diSessa argues that the field does not yet know which of the concepts from computing are most useful in a particular domain, but that this determination should be made based on computing's potential to reorganize and/or move a domain (or context) forward. diSessa therefore cautions against the essentialism: computing has no power on its own, nor does the ability to "think computationally" (if such a thing exists) enable any special cognitive capacities or "high-order thinking skills" that make one a better thinker more generally. So, whereas Steve Jobs famously asserted that "everybody…should learn to program a computer because it teaches you how to think," diSessa argues that computing knowledge is best treated as a literacy that "needs a literature" (i.e. many different contexts in which to represent ideas about the world) to have value and meaning. It is not the "infrastructural representational system" itself that is meaningful, but how this system enables people to represent, think about, and reorganize specific ideas in a specific context:

> Literacies shift the basic intellectual structure of domains of knowledge along with learning trajectories and societal participation structures–who gets to do what. Concomitantly, a literacy needs a literature. One needs to transcend a representational system by itself, and get to civilizations' expanse of deep and powerful ideas. Algebra has permeated almost all scientific and technological literature. It is taken for granted, but its value is distributed among many profound and not-so-profound "things civilization has managed to think and explain" using the representation, constituting a deep and rich literature (diSessa, 2018, p. 7).

The very idea of computational thinking should therefore be replaced by the notion of *computational literacy* – where computation is treated as a means to an end, where specific computing practices will necessarily vary from context to context (diSessa, 2018). This stands in contrast to *computational thinking*, which treats computing skills / dispositions as ends in themselves. These different views on "common knowledge" computing have profound implications for computing education: whereas in the (dominant) *computational thinking* model, any context will do, so long as it motivates the concepts enumerated above; the literacy perspective privileges the context itself, where computing is a tool and a representational system to make progress in a particular domain. These are critical epistemic differences that warrant considerable attention as curriculum and learning experiences continue to evolve in K-12 education.

1.1.2 Sociocultural and Sociopolitical Perspectives on Computing

Like diSessa, researchers who align with the sociocultural tradition also see context as an essential part of learning. However, the sociocultural tradition tends to take a more expansive view of context, with more emphasis on its cultural and political dimensions. From the sociocultural perspective, people develop their knowledge and practices within particular social environments, surrounded by other people, tools, social norms, and rules; and shaped by broader society.[5]

Much of the sociocultural work on computing and data literacy takes place within new media literacies research, filtering the "common knowledge computing" question through the lens of culture. Rather than seeing the goal of computing education as cultivating of a set of context-specific knowledge structures in the individual mind, new media literacies research considers learning as moving into new genres of participation and taking on new identities within a cultural context (Ito et al., 2010). Specifically, this line of inquiry examines how routine engagements with sociotechnical systems – i.e. young people's everyday cultural practices – shape important human endeavors, such as developing and maintaining friendships and familial bonds, participating in creative production, playing games, negotiating social hierarchies, finding information, and so forth (boyd, 2007; Gutierrez et al., 2017; Ito et al., 2010). It is in the pursuit of these fundamental human needs and desires that new literacies develop:

> Those immersed in new digital tools and networks are engaged in an unprecedented exploration of language, games, social interaction, problem solving, and self-directed activity that leads to diverse forms of learning. These diverse forms of learning are reflected in expressions of identity, how individuals express independence and creativity, and in their ability to learn, exercise judgment, and think systematically (Ito et al., 2010, p. xi).

Educational research within this literacy orientation has explored how young people's organic, everyday engagements might support particular learning objectives in formal spaces of education. Most of these efforts have explored the representational and communicative aspects of computing. For example, researchers have examined the kinds of stories and games that young people create with new computational media, and the learning opportunities these offer – in terms of language, communication, and coding skills (e.g., Peppler & Kafai, 2007).

[5] Discussed in more detail in chapter 2.

Efforts have also been made enhance the cultural relevance of the high school computing curriculum by more explicitly connecting computing lessons to familiar, everyday applications. For instance, the Exploring CS (Goode & Margolis, 2011) curriculum, which is intended to be taken prior to AP computer science, touches on ethics, HCI (human-computer interaction), the web, and some computing principles, in order to help learners understand the technical underpinnings of computing technologies that they regularly use. However, making these kinds of connections resonate with learners is not a sure thing. As Philip & Garcia (2017) and Sims (2017) both observed, appropriating students' texting, gaming, and social media practices into a formal curriculum does not guarantee engagement, and can even backfire (see T. Philip & Garcia, 2013 for discussion).

New media literacy research has also explored some of the sociopolitical applications of computing (e.g., C. H. Lee, 2012; Vakil, 2016). For instance, Clifford Lee (2012) offers the concept of *critical computational literacy* – using computer science concepts and skills "with the explicit purpose of creating sociopolitical awareness and ideological change by examining marginal perspectives and questioning commonly accepted beliefs" (C. H. Lee, 2012, p. 126). Lee examined this idea by noting how high school students leveraged the Scratch[6] platform to produce counter-narratives that challenge problematic stereotypes and ideologies. In the production of these computational critiques, Lee argues that students can be empowered to re-mediate broader deficit narratives, which are propagated across everyday media landscapes.

Similarly, the idea of *computational action* (Tissenbaum, Sheldon, & Abelson, 2019), which comes out of computer science, argues that computing ought to be presented as a way to take direct action in the world (often leveraging user-generated data). However this approach focuses more on the functional (versus representational) aspects of computing. Tissenbaum et al. (2019) describe the potential for data-driven apps to make an impact on the world (using the App Inventor platform developed by Abelson and his colleagues):

> Young learners have the capacity to develop computational products that have authentic impact in their lives from the moment they begin to code. They simply need contexts that allow them to have such impact (Tissenbaum et al., 2019, p. 36).

However, the computational action perspective also has its challenges. For one, this framing runs the risk of reducing important social problems (e.g. safety, environmental stewardship, etc.) to information and communication issues that can be solved by creating an app. In so doing, it subordinates important parts of the social, cultural, and historical dimensions of these "authentic, real-world" contexts, including the role of institutions, policies, political constituencies, and material opportunities. While computing can certainly be one means (of many) of achieving broader social goals, the "solutionist" (Morozov, 2013) approach that

[6] A blocks-based learning environment for making games and stories while exploring various computer programming concepts. See https://scratch.mit.edu/.

undergirds computing for "direct impact" – "from the moment [learners] begin to code" – creates its own internal contradictions (to be discussed in chapter 4), while at the same time producing an uncritical perspective of the role of computing in society. By treating social challenges as the context through which computational ideas can be explored, the problem domain becomes subordinate to the computational learning agenda; while the utility of a computational solution is taken for granted.

1.1.3 Data Modeling and Data-as-a-Service

Over the past 15 years, data-centric approaches to computing have become increasingly important within the field of computer science. Applications like machine learning, artificial intelligence, image recognition, and natural language processing – which permeate sociotechnical systems – are sought after in many industries and domains, and undergird many sociotechnical systems. Data science education research therefore offers another perspective on the practices, skills, and ideas needed to understand sociotechnical systems. In this dissertation, I use Philip, Schuler-Brown, and Way's definition of *data science* as the "interdisciplinary and multidisciplinary approaches to the study, use, and application of data" (2013, p. 114). I also define *data literacy* as the extent to which someone is able to participate in the practices of data science. From my perspective, research in data science education considers two distinct, but related data practices: (1) *data modeling* – a term that describes how data are used within various knowledge production processes; and (2) *data-as-a-service* – which treats data as a kind of atomic material that can be transmitted, assembled, replicated, etc. across networks of disparate systems of meaning.

Data Modeling

The dominant way of conceptualizing data literacy[7] in the K-12 data education research involves the practice of what Lehrer (2007) calls *data modeling* – the iterative process of engaging learners in generating and posing questions, selecting attributes and constructing measures, structuring and representing data, and drawing inferences. Philip, Schuler-Brown, and Way describe a similar process of "formulating questions and engaging in the generation, collection, representation, visualization, analysis, interpretation, and communication of data" (2013, p. 114). From this perspective, data literacy enables an empirical, systematic knowledge production process involving deconstructing real world phenomena into attributes and measures, and aggregating/analyzing these measures across many instances in order to characterize its size/scale or describe relationships between phenomena.

Data modeling is typically done within a traditional disciplinary context (which aligns with the computational literacy camp) – as concepts and their relations must be understood within a framework or theory of the world. Many learning researchers have argued that data can make abstract ideas more tangible and relatable. For instance, in mathematics education, Ben-zvi &

[7] I use the term *data literacy* to refer to the ability to participate in the practices of data science.

Arcavi (2001) argue that data provide a way to concretely explore mathematical formalisms by collecting and visualizing observable aspects of one's environment. Similarly, in social studies and in environmental science education, data modeling has been used as a way to help learners examine larger social and ecological patterns, such as residential segregation (e.g., Elwood & Mitchell, 2013; Enyedy & Mukhopadhyay, 2007); or access to financial institutions segregation (e.g., Rubel, Hall-Wieckert, & Lim, 2017) – where data represent real, observable phenomena that learners can more easily reason about. As computational tools have become more accessible in K-12 educational settings, some research has also explored how computing might be more integrated with data science education – moving towards a more contemporary view of data science at the intersection of statistics, computing skills, and domain expertise (see Figure 1). Examples of this include helping young people to analyze data from sensors and wearables (e.g., V. Lee & Drake, 2013), or easing learners into working with statistical software (like R) to leverage more complex data-related computations (e.g., McNamara, 2015).

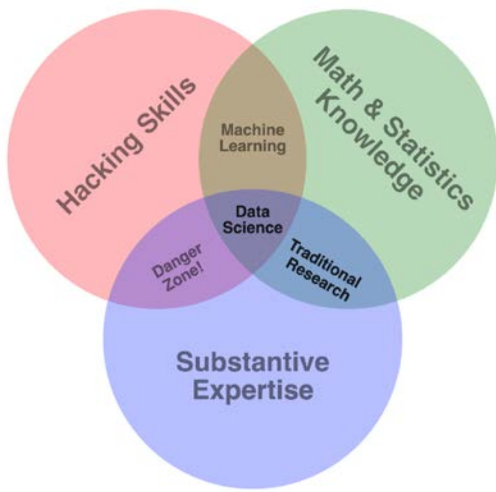


Figure 1: Drew Conway's Data Science Venn Diagram (Conway, 2010)

Many data education researchers also argue that data literacy involves understanding the social implications of data. For instance, Philip, Schuler-Brown, and Way (2013) argue that a data literacy framework should help learners consider the ways in which measures and categories do real work in the world – which involves considering the assumptions imbued in data, and how data science might be leveraged to both challenge and reinscribe power (elaborated in chapter 3). Learners must also consider the goals to which data science are directed (e.g. reinscribing stereotypes versus understanding and potentially addressing important social problems); the conclusions that can and cannot be drawn from data; and the kinds of policies and decisions (the good and the bad) that have been advanced via data-driven knowledge production. As data-mediated knowledge production continues to expand and develop, supporting people in reasoning about the social implications of data has become increasingly important. That said, how this criticality might be fostered is still very much an open question.

Data-as-a-Service

A second way of thinking about data literacy considers data as a raw material whose value is contingent on the sociotechnical infrastructures that create and leverage it. While the data modeling process tends to imply a static notion of a dataset – a fixed snapshot of entities and attributes that can be analyzed and visualized to inform practice – many modern systems leverage data which are constantly being created, updated, shared, replicated, transformed, filtered, archived, bought-and-sold, and passed onto other systems. Moreover, these distributed, networked, data-driven computational systems are likely the context that Wing was thinking of when she coined the phrase "computational thinking" (shown in section 1.1.1), which diSessa (2018) gestures to.[8]

While these systems have enabled wonderful new conveniences – accessing real-time traffic reports, scheduling an Uber at the click of a button, checking the Mets/Giants score from your Uber, etc. – they have also introduced new risks, including to our online privacy and security. Therefore, new critical dispositions are also needed to make sense of these systems, and consider some of the ways in which they might be problematic. Whereas *critical computational literacy* considers computer science concepts as a means to "generate sociopolitical awareness" (from social issue to computational solution), these same critical dispositions are also needed to critique the systems that are already with us (from computational artifact/system to social/policy/institutional solution), and how they shape important social values regarding consent, transparency, autonomy, safety, and others. This might include helping learners to:

- Recognize the *sites* of data collection, including the who, what, when, where, why of data construction and collection.

- Understand how various sources of data generate value/revenue.

- Consider the pros and cons of personalized, data-driven algorithms, which have been trained via data gathered about individuals and their networks. How might these shape the kinds of news you read, the kinds of information that get produced / not produced and found / not found (e.g. filter bubbles, silos, echo chambers, etc.)?

- Consider what rights an individual might have regarding the data that has been collected about them, including the right to be forgotten and the right to opt-out.

[8] Many computational thinking concepts – patterns, models and simulations; systematic processing of information; algorithmic notions of flow of control; and so forth – are particularly useful for building and reasoning about these systems.

1.1.4 Implications for Sociotechnical Literacy

To summarize, each of the ideas reviewed offers important considerations regarding the kinds of computing- and data-related knowledge that are important for reasoning about the applications of computing; and some ways of thinking about context. In light of this literature, I argue that as young people engage with these technical ideas, considerations of context are essential. Young people will learn the ideas that are motivated by the contexts in which their learning takes place. While making games and animations will motivate considerations about events, loops, and game mechanics, these activities will not necessarily motivate learners to examine some of the possibilities, limits, and risks of the information and communication infrastructures that shape their lives (e.g. Google, Facebook, YouTube). Having some baseline technical knowledge can certainly help young people to reason about the mechanisms through which these systems operate. However technical knowledge must also engage with some of the everyday sociotechnical applications that surround us, so that learners can understand *how* computing mediates human activity in particular circumstances, subject to particular constraints and possibilities.

If the goal of computing education is to produce citizens[9] with robust computing and data literacies (as opposed to producing tech professionals and software engineers), how might it change what is considered to be fundamental within computing and data education, and the contexts in which computing and data science are situated? Perhaps rather than centering computational thinking's "universal abstractions" – which appear to have been lifted out of computer science – it might be prudent to consider some of the other ways in which the contexts of computing and data have been conceptualized (e.g. within data, media, and literacy education). Whereas computational thinking certainly encompasses important computing ideas, a literacy perspective views computing and data in terms of its ability to help people interpret and participate in the world.

A literacy perspective also expands the field of view by embracing some of the social and political aspects of data and computing. This includes an interrogation of the very categories, attributes, entities, and relationships that are taken for granted within data science practice (to be examined in chapter 3) and the kinds of benefits that are often attributed to data-driven computing. This might also include looking at the possibilities, limits, and risks of data infrastructures (e.g., Jonathan Gray, Gerlitz, & Bounegru, 2018; T. M. Philip et al., 2013), such as how information circulates, the monetary incentives that impact these infrastructures, and the magnitude and scale of personally identifiable information that is available on the Internet.

---

[9] I use the term "citizen" to refer to the idea of being an active participant in the social, political, and institutional infrastructures that frame one's life, and the lives of one's community. I am not referring to their legal status.

1.2 CSforAll

There is little doubt that the demand for computing and data education in K-12 has grown steadily over the past decade amongst parents, teachers, school administrators, and industry (Google Inc. & Gallup Inc., 2016). By and large, this demand has been driven by the belief that a robust computing education in K-12 is essential for positioning the students of today for the high-opportunity, computing-intensive careers of tomorrow. Schools therefore have an obligation to arm students with these emerging 21st century literacies. As one example of this rationale, consider the following excerpts taken from a 2013 viral marketing video, entitled "What Most Schools Don't Teach (Code.org, 2013)" produced by Silicon Valley Backed Code.org, the nation's preeminent *CSforAll* advocacy organization:

> Whether you're trying to make a lot of money, or whether you just want to change the world, computer programming is an incredibly empowering skill to learn.
> – Hadi Partovi (co-founder of Code.org)
>
> The programmers of tomorrow are the wizards of the future. You know, you're going to look like you have magic powers compared to everybody else.
> – Gabe Newell (founder of Valve, a game development/platform company)
>
> I think if someone had told me that software is really about humanity, that it's really about helping people by using computer technology, it would have changed my outlook a lot earlier.
> – Unnamed white woman
>
> Great coders are today's rockstars.
> – Will.i.am (creator of the Black Eyed Peas)

The celebrities in the video (Mark Zuckerberg, Bill Gates, NBA AllStar Chris Bosh, and pop star Will.i.am) describe a long list of enticing possibilities that are enabled by learning to code: making a lot of money, helping people, changing the world, or becoming "a wizard of the future." However, at the close of the video, we learn that "one million of the best jobs in America may go unfilled because only 1 in 4 schools teach computer science." The video suggests that if schools fail to act, most children will be robbed of these future opportunities. This particular *CSforAll* narrative is interleaved into most computing education initiatives: for those who have *access* to a computing education, a litany of opportunities awaits, including a lucrative career that can change the world for the better.

Given these value propositions, states and school districts have scrambled to make computing education available to their students, and in doing so have grappled with many questions. Some of these are more logistical: should programming be a graduation requirement? Should it count as a foreign language? Who will teach it? How will we pay for it (e.g., Schneider, 2018)? Others are more political: are public schools granting the tech industry too much power to shape curriculum requirements (Singer, 2017)? How do we ensure equity in access and outcomes across gender, race, socio-economic status, and geographic region (A. Scott, Martin, McAlear,

& Madkins, 2016)? When the social and political contexts of computing are examined within educational efforts, they are primarily viewed through the lens of access: whether access to high quality computing education is evenly distributed across social groups. However, it is rare that the contexts of computing themselves are questioned, including the ethical dimensions of computing technologies, or whether and how these technologies shape different groups of people's life chances.

1.2.1 Equity and Access to Computing Education

Women/girls, African Americans, Latinx, and Native populations are disproportionately underrepresented in computing, both in computing classes and in the computing field more broadly (Margolis, 2008). This of course is troubling, given the increasing prevalence of computing- and data-mediated systems. As a result, there has been a coordinated effort to understand and address barriers to participation. Researchers have identified several structural, social, and psychological barriers (A. Scott et al., 2016). One structural challenge is that schools that predominantly serve low income students and students of color are less likely to offer high quality computing courses than their wealthier, whiter counterparts (Margolis, 2008). However issues surrounding underrepresentation extend well beyond access to courses. Other issues include students' perceptions of the relevance of computing to their lives (C. H. Lee, 2012; Ryoo, Margolis, Lee, Sandoval, & Goode, 2013), beliefs around whether they are likely to benefit from computing knowledge, implicit bias (Cheryan, Master, & Meltzoff, 2015) and stereotypes (Steele & Aronson, 1995) within and outside of the classroom, and whether students see people from their own social or cultural communities doing computing-related work (Anderson, Lankshear, Timms, & Courtney, 2008; Cain, 2012).

Many efforts have been launched across the computing education pipeline in order to broaden participation. Some of these are organizational, such as creating academic and industry conferences geared towards women and underrepresented groups, and hosting various research summits sponsored by the ACM (Association of Computing Machinery) and NSF (National Science Foundation) to examine the diversity question (e.g., Cooper, Grover, Guzdial, & Simon, 2014). Other efforts involve curricular and pedagogical interventions: situating the abstractions of computing in contexts that are meaningful to underrepresented minorities and girls (e.g., K. a. Searle, Fields, Lui, & Kafai, 2014); and encouraging culturally relevant (Ladson-Billings, 1995) approaches to teaching and learning about computational thinking (Rorrer, Allen, & Zuo, 2018). Others involve disrupting the idea of the "geek gene" – the idea that some people (typically people from dominant social groups) have the predisposition for thinking computationally (Guzdial, 2016).

However, while equitable access to computing education is an important and worthwhile goal, it is but one aspect of the equity question within the broader landscape of data and computing technologies.[10] Not everyone who has access to a computing education can or will go on to

[10] I define equity as (a) taking into account both the opportunities and constraints that are made available across and within social groups, given a set of circumstances; and then (b) ensuring that the

pursue a tech job, nor would society function if they did. Moreover, accessing high-opportunity careers depends of other litany of factors: various forms of social capital, access to high-opportunity networks and/or knowing someone in the industry, a fair and unbiased hiring process, and so forth. Therefore, having the opportunity to learn about the abstractions of computer and data science is not the same thing as having the opportunity to benefit equally from knowing these abstractions. As such, learning about computing is not the same thing as fostering a sociotechnical ecosystem which benefits all social groups equally.

1.2.2 Computing Applications and Equity

A jobs-centric view of computing and equity also ignores some of the broader, computer-mediated applications and trends that work against historically non-dominant social groups and the poor. This includes the automation of low-wage work, the diminishing employment benefits (healthcare, retirement) that have accompanied the "gig economy," the expansion of surveillance technologies (Eubanks, 2018), and the amplification of bias and stereotypes in information services (Noble, 2018). That said, within the *CSforAll* rhetoric, computing- and data-intensive technologies are described almost exclusively in terms of their potential social benefits, reflecting a broader utopian ethos that has a long history within the tech industry (see Turner, 2006 for extended discussion).[11]

In reality, new computing innovations have both benefits and limitations. For instance, Uber directs revenue away from the taxi cab industry and towards itself, its "gig drivers," and its shareholders. However, rides are cheaper and more convenient for the consumer. Similarly, Google and Facebook's respective advertising platforms have crippled the ability for newspapers to sell advertisements (thus undercutting the viability of journalism itself), but have also funded people's ability to use search engines and social media sites for free,[12] which also has tremendous value. Civic and open data initiatives have (sometimes) made aspects of government more transparent; but have also made it much easier for (larger) businesses with substantive technical expertise to leverage these datasets to target vulnerable people. For instance, during the housing crisis of 2008, large real estate agencies leveraged a foreclosure database in order to more efficiently target and then convince vulnerable homeowners in

---

sum of opportunities (a positive) and constraints (a negative) is equal. This perspective on equity means that those who experience more constraints (which are typically historically produced and cumulative over time) are provided with more resources (and vice versa).

[11] For instance, Facebook is not providing the infrastructure for all forms of information and communication activities – the good and the bad. It is "giv[ing] people the power to build community and bring the world closer together" (Zuckerberg, 2017). Uber is not undercutting the cab industry and paying drivers sub-minimum wages (Mishel, 2018). It is providing "transportation as reliable as running water, everywhere for everyone."

[12] By "free", I mean that users do not have to pay with money, but they do pay with their data, their time, and their clicks.

Oakland, CA (and nationwide) to sell their homes at a fraction of their value. These firms were then able to "flip" these homes a few years later and sell them at a massive profit (Urban Strategies Council, 2012), thus displacing thousands of residents.

While education may not be able to address some of the problematic implications of data- and computing- technologies (or at least not in the near term), it can offer a more balanced treatment of the applications of these technologies. Ladson-Billings (1995), who coined the phrase "culturally relevant pedagogy," argues that for educational initiatives to work for all students (thereby fostering equity), they must be "culturally relevant," and to be culturally relevant, they must meet three criteria: "an ability to develop students academically, a willingness to nurture and support cultural competence, and the development of a sociopolitical or critical consciousness" (Ladson-Billings, 1995, p. 483). The idea of cultural relevance is often deployed as a way to engage students from non-dominant social groups with the abstractions of computing, by connecting students' interests (music, art, etc.) to various computing- and data-related activities. However, leveraging students' cultural practices to motivate skill development misses the spirit of culturally relevant pedagogy, namely acknowledging social ills and inequities, and arming students with the tools to navigate the (unevenly distributed) opportunities and constraints that they might encounter. I argue that examining sociotechnical systems offers a context that can enable learners to engage with rigorous "academic" ideas while honoring students' cultural worlds *and* examining some of the sociopolitical realities that frame young people's lives.

1.3 Organization of the Dissertation

In an effort to help young people understand how data and computing facilitate important human activities, this dissertation explores two approaches to situating technical ideas within real sociotechnical systems. In the first sociotechnical system, young people, ages 15-22, engaged in two different neighborhood initiatives in order to advocate for targeted public investments to foster community health and well-being. This involved gathering and analyzing data to characterize local assets and challenges, and wrestling with some of the politics of public resource management. In the second sociotechnical system, high school students examined some of the social media platforms that they regularly used. This involved learning about the possibilities inherent to secondary uses of social media data, building various third-party apps, and discussing some of the privacy and economic implications of platforms. Both approaches aimed to make the *applications* of computing primary (rather than treating applications as the backdrop from which the abstractions of computation are motivated), so that learners could examine some of the specific ways in which data and computing might be directed to particular goals, subject to real possibilities and constraints, and in relation to alternative forms of participation.

The dissertation is organized as follows: in chapter 2, I situate the empirical work presented in this dissertation within theories of learning and context. I also introduce the methodological approach that I used to explore the problem space, which includes a reflection on my own thinking about computing- and data-education and equity at the beginning of this research project. While I came into the research with a commitment to the very same utopian ideals as those described in the *CSforAll* section, the design experiments helped me to see the limits and contingencies of computer-mediated systems – particularly for people from more vulnerable social groups.

In chapter 3, I consider what sociotechnical literacy development might look like in the context of data-driven civic advocacy, as it relates to the management of public resources (e.g. including air and water, schools, public spaces, affordable housing). Managing public resources is technically, politically, and logistically complex, and is increasingly mediated by data-intensive processes that have a profound impact on individual and community wellbeing. As such, this context offered young people a situated way to learn about data, and to examine *whether* and how data-centric ways of knowing figure into broader civic efforts. Within the chapter, I describe how each focal project unfolded, including how the projects were framed; the data scientific practices and ideas (mostly around data modeling) that students engaged with/in; students' perspectives on what they were working towards; and what they managed to achieve (and not achieve) through their respective data inquiries. Using transcripts and student-generated artifacts, I argue that a holistic treatment of data (i.e. where data served as one means of pursuing broader goals) pushed students to consider whether and how their data-related efforts fit into their broader goals related to advocacy and activism. Given that data-driven knowledge production is often described in terms of its ability to inform more principled, fair decision-making, these projects helped young people challenge this utopian ideal (absent other political and institutional mechanisms), while also showing them some of the strengths of engaging collective, systematic inquiry using data – fostering solidarity, confidence, and a partially shared political identity among students facing a common challenge.

In chapter 4, a transitional chapter, I describe a series of subsequent pilot studies that aimed to address some of the tensions that emerged in the civic advocacy study – namely, (a) providing students with more choice and agency in selecting and pursuing their ideas; and (b) trying to direct students towards goals that were more attainable. These pilots were essentially efforts to discover whether there might be better ways to bring computation and data to bear on civic advocacy that could have a real impact. After exploring these approaches, with limited success (if civic action was the goal), I decided to shift my focus to some of the other ways in which high school students were already participating, organically, in other "real world" sociotechnical systems, of their own volition. These learnings became the basis for a second design experiment, described in chapters 5-7.

In chapter 5, I use findings from these pilot studies and related work in computing education to describe the rationale for a second design experiment, which engaged high school students in the study of networked platforms. I explain how I instantiated this design experiment through a set of learning goals, tools, and activities; and describe how these relate to other educational initiatives to teach about computing in context.

In chapters 6 and 7, I describe some key findings from this design experiment, and the resulting implications for teaching and learning about networked platforms. In chapter 6, I attend to some of the computing knowledge, skills, and dispositions that students were practicing as they worked towards their various project goals. Here, I note that learning to read and interpret domain-specific declarative languages and data formats, such as HTML, CSS, and JSON, can offer students important insights into computing that are not as easily accomplished in a traditional programming progression, because they are designed with the intent of querying, presenting, and manipulating networked content. I also attend to some of the questions, confusions, and curiosities that students expressed regarding the more social and economic aspects of the web. I show that while students generally knew that issues of privacy, security, and money were pertinent to their everyday interactions with technology platforms, they didn't necessarily understand the underlying mechanisms in play. These findings highlight some important distinctions between *knowing that* and *knowing how* platforms leverage user-generated data, and suggest that supporting inquiries that give more insight into "the how" might enable students to more carefully reason about (and potentially question) the mechanisms and risks that undergird how content is gathered and distributed.

In chapter 7, I draw from student interviews, discussions, and trajectories of participation during their 5-week "internship," to consider how they understood the computing technologies in the context of their lives, and their motivations for seeking out computing knowledge. Understanding students' prior experiences and goals contributes to growing theory regarding how young people are making sense of their computationally mediated world.

Finally, in chapter 8, I reflect on what these studies collectively reveal about the possibilities, limitations, and risks of data and computing, as situated in the lives of young people; as well as what this might mean for helping young people develop a robust sociotechnical literacy. There are very real limits to what can be accomplished with computing and data alone. There are also significant benefits *and risks* associated with the many sociotechnical systems that shape our lives. As such, I argue that rather than positioning computing education as a remedy to various social ills, we instead offer young people a fair explanation of what computing is and is not capable of, grounded within specific contexts involving real people. I close with what this fair explanation might include, and how it might be fostered.

2 THEORY AND METHODS

In the previous chapter, I consider what might constitute a sociotechnical literacy, and how this idea relates to other ways of thinking about computing- and data-related knowledge. In this chapter I describe some of the concepts that guided the design and analysis of the empirical work presented in this dissertation. I begin by reviewing theories on (a) learning and (b) the factors that shape learning (i.e. context), and locating my own conceptual commitments within these theories. I then describe my methodology and reflect on my own positionality and biases that I brought into the work. I conclude by summarizing the chronology of the research and what I was able to learn from this kind of longitudinal, iterative empirical approach.

2.1 Perspectives on Learning

In order to situate my own view of learning, I first briefly summarize three of the most prevalent theories of learning: behaviorism, cognitive constructivism, and social constructivism. Each theory views knowledge and motivation differently, which in turn inform very different approaches to organizing educational experiences. The earliest of these theories, *behaviorism*, came out of work by B.F. Skinner, Edward Thorndike, and John B. Watson and others in the early 20th century. Behaviorists view knowledge as a set of learned responses to stimuli in the world, which can be shaped through positive and negative reinforcement. From this perspective, the role of educational spaces is to reinforce appropriate responses to stimuli, which are passively absorbed by the learner through practice and repetition. As modern schools were developed and organized during the time that behaviorism was prominent, much of the current organization of schooling is reminiscent of this philosophy of learning – where important ideas are abstracted into general concepts and algorithms, and drilled in order to teach "the basics" (A summary of the behaviorist theory of learning can be found in chapter 1 of Wenger, 1998).

Cognitive constructivism, which emerged in the early- to mid-twentieth century, challenged the behaviorist notion of people as empty vessels in need of facts and stimuli. Jean Piaget, perhaps the most influential thinker from this camp (Ginsburg & Opper, 1988), formulated a developmental theory of knowledge that described how people actively construct knowledge as they experience the world through the processes of (a) assimilation – fitting their experiences of the world into their existing mental structures, and (b) accommodation – re-configuring their mental structures to come to more parsimonious and coherent understandings of the world (Ginsburg & Opper, 1988). By observing and writing about his own three children, he showed that children have many ideas, and a coherent framework for thinking about how the world works. However, children's frameworks are qualitatively different from adults, following a fairly regular series of universal, developmental stages. A key feature of Piaget's constructivist theory is the importance of the individual child and her experience. In Piaget's model of learning, children develop based on their individual, phenomenological experiences: although the social world can influence one's understanding of perspective,[13] it is not a central process of development. Moreover, in contrast to behaviorism, the cognitivist see motivation as something that is mostly intrinsic, based on an individual's interests and curiosities. Educational spaces that are organized according to this perspective tend to treat learning as the active process of discovery where learners have the option of directing their own learning through inquiry and a variety of tools and materials.

A third perspective, *social constructivism*, also known as sociocultural theory, emerged from a group of soviet psychologists in the early/mid twentieth century. This theory questioned the cognitivist assumption that learning can be adequately described as an individual, cognitive process. The most famous of these psychologists, Lev Vygotsky, argued that language, culture, and social relations shape how people interpret the world. Meaning is not derived from scratch based on biological hardcoding, but culturally imposed on one's sensory perceptions from a learner's social world – rich with language, tools, and artifacts. As Vygotsky explains:

> I do not see the world simply in color and shape but also as a world with sense and meaning. I do not merely see something round and black with two hands; I see a clock. (L. S. Vygotsky, 1967, pp. 16–17)

The idea of "a clock" – an artifact of the social world – precedes the learner, as does the meaning associated with the information the two hands represent (e.g. *I'm late! Almost dinner time!*). From this perspective learning and development exist first at the cultural level (surrounded by language, artifacts, and social arrangements that are already infused with meaning), and then at the individual, psychological level. From this perspective, learning is a deeply social process, whereby people learn through interactions with their cultural communities and environments, in order to more fully participate in those communities and environments. Rather than attributing a person's visions, perspectives, interpretations, and choices to their own individual mind, sociocultural theory examines the ways in which a culturally-mediated environment – filled with artifacts, rules, norms, values, and social interactions – shapes how a person understands and acts in the world. As such, motivation is both intrinsic and extrinsic; and learning is viewed as the extent to which the learner is able to participate in a valued social practice, versus via the acquisition of concepts and mental representations (Greeno, 1997). Over the past two decades, sociocultural theory has also become more attentive to the political dimensions of learning, and how power also operates within places of learning, and in the larger cultural plane. This includes looking at the ways in which power and social position (e.g. race, class, gender, etc.) mediate how opportunities and resources – both material and ideal – are offered to different categories of people, such as

[13] Perspectivism: 'my understanding / experience of X is different from yours.' For example, one of the tests Piaget uses to understand a child's understanding of perspective is to ask a child "what the doll sees" from various perspectives on a model landscape. A young child always reports her own perspective, despite the experimenter's variations in the doll's position.

socio-economic class (e.g., Freire, 1996; Willis, 1977), race (e.g., Gutiérrez & Rogoff, 2003; Gutiérrez & Vossoughi, 2010; Nasir & Hand, 2008), or gender (e.g., Langer-Osuna, 2011).

Educational spaces that embody this particular view of learning may look similar to those informed by the cognitivist perspective. However, because learning is viewed as developing fluency in a cultural practice, more attention might be given to collective learning across the group, including the trajectories of participation that each member takes within the practices, or how various roles work together relationally. In addition, because culture is imbued in the tools, norms, rules, etc. of the cultural practice, each of these mediators is treated as a carrier of cultural knowledge, and hence more carefully interrogated (at least in theory). In other words, rather than treating a particular learning practice as neutral, sociocultural theory asks whose practices and knowledge are being passed onto a learner, and towards what goals. For the purposes of this dissertation, I use the term "the situated perspective" to refer to the social constructivist view on learning.

2.1.1 The Cognitive / Situated Learning Debate

Since Piaget and Vygotsky first formulated their respective cognitive and situated theories of learning, each theory has evolved and contributed substantially to the emerging discipline of the *Learning Sciences*. That said, key differences remain between the two theories – in terms of how each conceptualizes knowledge, learning, and activity (summarized in Table 1, and adapted from Greeno, 1997) – which have important implications for how a sociotechnical learning efforts might be organized and understood. Greeno (1997) summarizes some of these differences: from the cognitivist perspective, knowledge is conceptualized as a structure that exists in the mind of the knower; and learning is the process through which an individual develops concepts, beliefs, theories, schema, structures (many different terms are used and analyzed), and how these mental structures generalizes to novel situations – in a process known as "transfer." From this point of view, key lines of inquiry include identifying what these abstract concepts are, how people develop them, and whether and how how they generalize to new contexts and domains.

On the other hand, the situated perspective conceptualizes knowledge as a relational, dynamic process within socially situated activity, which Greeno defines as the "properties of interaction involved in the coordination of behaviors of the various system components in ways that enable successful participation" (Greeno, 1997, p. 8). To give an example of what this situated view of cognition, Lave (1988), in her study of everyday mathematics, shows that even something as presumably universal as *doing arithmetic* looks qualitatively different across different contexts (e.g. finding deals at the grocery store; counting calories for weight loss) – depending on the subject's, goals, surrounding resources, and the relations between them. Moreover, people perform everyday mathematical tasks much more easily than (supposedly) analogous school tasks, precisely because the purpose for doing mathematics is clear; and because familiar resources offer assistance in performing these calculations. In the real world, problems are encountered in situ versus imposed via curriculum (Packer, 2001). Given this view of thinking and doing, learning is defined as the process through which a person develops the practices needed to participate in a socially situated activity, given the resources around them. Some of the inquiries taken up in situated learning theory include understanding: how and why people take on new roles and identities within a social practice; and how different aspects of a practice (artifacts, tools, social arrangements, cultural norms, and so forth) support successful participation in it.

Table 1: Comparison of Cognitive and Situated Perspectives (Greeno, 1997)

	Cognitive	Situated
Unit of analysis	Processes and structures that are assumed to function at the level of individual agents	"Participants, interacting with each other and with material and representational systems" (p. 7)
Knowledge	Properties of an individual's mind. Conceptualized as a substance or structure attributable to the knower	"Properties of interaction involved in the coordination of behaviors of the various system components in ways that enable successful participation" (p. 8) Conceptualized as a process of "knowing" – regular patterns in someone's participation in interactions with other people and with material and representational systems.
Learning	The process through which an individual develops the mental structures that constitute knowledge	The process through which a person develops the practices needed to take on meaningful roles within a complex, social activity.
Generality of Learning	Acquiring abstract representations as well as procedures for applying them to many situations	Acquiring repertoires of practice that succeed over a broad range of situations; which can in turn contribute to an individual's identity as a valuable participant in those activities
Assessment	Test various concepts in an individual's mind	Requires sampling of certain practices that a person has acquired across a range of situation types

One of the key fault lines in the cognitivist and situated perspectives involves the idea of "transfer," or more broadly, the generalizability of knowledge. "Transfer" is a concept within the cognitivist camp, and refers to the idea that a person who acquires knowledge from one context should be able to apply it to similar contexts in the future. Transfer is a normative idea in educational discourse (including in technical education – e.g. "the concept of iteration does not easily transfer from Scratch to Java"). However, situated theorists have critiqued the notion that knowledge is a collection of:

> 'tools' for thinking, 'appropriated' into the toolkit of mind, transported from one situation to another, and applied to new tasks. The tools are assumed to be independent of the situations in which they are used, unchanged as they are taken from one task to the next" (Lave, 1988, as cited in Packer, 2001, p. 498).

Instead, the situated perspective describes the generality of learning in terms of repertoires of practices – whether an individual has acquired the skills and ways of knowing and doing that succeed over a broad range of situations; which can in turn contribute to an individual's identity as a valuable participant in those situated activities (Greeno, 1997).

Like Vygotsky, Lave, Greeno, and others, I also see learning as situated and contingent on particular activities, people, tools, and social arrangements (i.e., "context"). This means that, for instance, doing data science in the context of neighborhood planning motivates a particular set of practices and activities that are qualitatively different than doing data science in the context of air quality monitoring, or making a data-driven app. While some skills and ways of knowing and doing generalize, many have ties to the particular goals that exist in particular contexts. Furthermore, if the goal is primary and the technical practices are a means to an end (versus the other way around), the learner actually has the opportunity to assess the purpose of a technical practice, and whether it even makes sense to use such a practice vis-à-vis alternate forms of participation. This stands in contrast to making, for instance, "data modeling concepts" primary (constructing measures, making inferences, interpreting graphs), and then putting each "concept" into context to ground it. While this exercise might make sense within a larger social practice, say, *making a data visualization to show at a City Council meeting*, without a broader reason for making such a chart, the context becomes simply "doing schoolwork."

2.2 What is a Context?

Because the situated perspective views learning and context as inseparable, I briefly review how I define context, drawing on ideas from cultural historical activity theory and social practice theory (two situated theories of learning). In my view, both theories are compatible ways of characterizing how people, learn, act and find meaning in their activities, and each offer useful concepts for understanding the role of culture, social relations, and artifacts within social settings.

Cultural historical activity theory (CHAT), which comes out of the work of Vygotsky (Cole & Engeström, 2007), takes "activity" as its unit of analysis, where activity is defined as a group of people working together to achieve a common goal or objective. Within an activity system, CHAT offers a way of examining how various forms of mediation – tools/artifacts, norms/rules, and the division of labor – help actors to achieve their objectives and to resolve challenges. In the classic Activity Theory diagram below (Figure 2), the arrows indicate how the mediators of the system (located at the vertices), the "subject" (person) and the community interact and mutually shape one another as they work to achieve a a goal (i.e. "object").

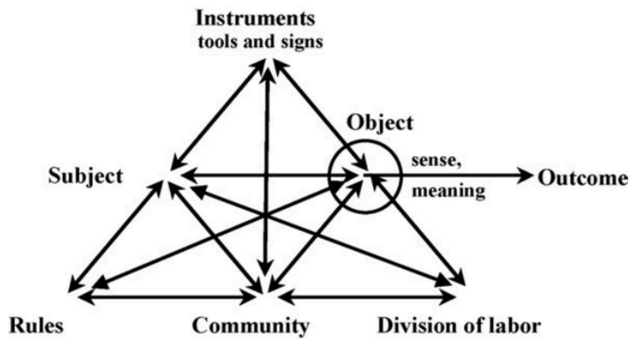


Figure 2: Schematic of an Activity System (Engeström & Sannino, 2010, p. 6).

Social practice theory, which derives from studies of apprenticeship communities (Lave & Wenger, 1991) and workplace learning (Wenger, 1998), also involves making sense of socially organized activities, but uses different terms and a different unit of analysis. This theory evolved out of work by Jean Lave, a social anthropologist, and Etienne Wenger, a learning theorist, as they examined a variety of "everyday learning" contexts – involving Liberian tailors, insurance claims processors, Weight Watchers meetup groups, and so forth. Within social practice theory, the "community of practice" is the central unit of analysis, which Wenger defines as the kinds of participation repertoires and social relations that are "created over time by the sustained pursuit of a shared enterprise" (p. 45). Communities of practice differ from activity systems in that they are focused on how practice-oriented communities sustain, reproduce, and reinvent themselves over time; versus how people come together to achieve a particular goal. The idea of communities of practice was formulated, in part, as a critique of the formalisms of school and behaviorism more generally (Lave 1988), in order to better account for the consequentiality of learning in everyday enterprises (work, family life, and so forth). However, I argue that ideas around meaning, identity, and belonging – central tenets of social practice theory – are very much relevant to organizing and understanding learning environments within formal settings as well (elaborated in chapter 5).

Both CHAT and social practice theory have similar commitments to the pursuit of meaning, and to the social and cultural dimensions of learning. Some of the similarities and distinctions of the two theories are summarized in Table 2, which was created by Wenger-Trayner (2013) to argue for the utility of assembling compatible social theories together to make sense of the social world. I use concepts from each theory to analyze two interrelated phenomena:

- How ideas (norms, values, and beliefs about the world) become embedded in social settings (e.g. learning environments) through the *tools* (artifacts and curriculum) students are asked to use, and the *goals* to which they are directed (using concepts from cultural historical activity theory).

- How people participate in social settings (e.g. learning environments) in light of their own pursuits of *meaning* and *identity* (using concepts from social practice theory).

Table 2: Distinctions and Points of Connection Between Activity Theory and Communities of Practice (adapted from Wenger-Trayner, 2013, p. 7)

Aspects	Activity Theory	Social Practice Theory	Compatibilities
Perspective	Context of purposefulness in which a practice is applied	Learning continuity across time and activities	An activity involves multiple practices (through the division of labor) and a practice is realized in multiple activities.
Joint enterprise	Object	Domain	Domain refers to areas of competence necessary for achieving the object of activities
Person	Subject	Identity (participation)	Identity refers to the continuity of learning and becoming across contexts of subject-object relation
Things	Tool (mediation)	Artifact (reification)	A tool mediating an activity is usually an artifact that has meaning within one or more practices (beyond a single activity)
Drivers	Good for locating contradictions that drive development	Good at focusing on learning opportunities in trajectories through the social landscape	Recognizing the role of multiple drivers of learning, including contradictions, but also participation, boundary processes, inspiration, adoption, etc.

2.2.1 Meaning and Identity

Identity is is a first-order concept in social practice theory (though implicit in CHAT), and concerns whether and how an individual is able to negotiate meaning and belonging within a community of practice. As Wenger explains, *identity* inextricably links the individual and her social world, so that each is defined in terms of the other. As he explains:

> [Identity] avoids a simplistic individual-society dichotomy without doing away with the distinction.…It does justice to the lived experience of identity while recognizing its social character – it is the social, the cultural, the historical with a human face.
>
> Talking about identity in social terms is not denying individuality but viewing the very definition of individuality as something that is part of the practices of specific communities.…Our practices, our languages, our artifacts, and our world views all reflect our social relations. Even our most private thoughts make use of concepts, images, and perspectives that we understand through our participation in social communities (p. 145-46).

Identity links the individual and the social by considering communities and social categories in terms of an individual's sense of belonging; and brings into focus the impacts that different kinds of social process (e.g. economic structures, power, social norms, etc.) might have on an individual (and vice versa). From this perspective, learning is the result of taking on new identities – as this process necessarily involves developing the kinds of competencies needed to occupy a new identity. As such, Wenger argues that education should be in the business of helping people to explore new identities within a practice, rather than solely focusing on "skill-building and information delivery activities (which tend to be highlighted in K-12 educational settings):

> Whereas training aims to create an inbound trajectory targeted at competence in a specific practice, education must strive to open new dimensions for the negotiation of the self. It places students on an outbound trajectory toward a broad field of possible identities. Education is not merely formative – it is transformative (p. 263).

2.2.2 Goals

While the pursuit of meaning and identity are the primary motivators that drive learning in social practice theory, CHAT puts a greater emphasis on particular goals, or "the object". In CHAT, goals are the primary way of defining the activity system's purpose. Furthermore, goals[14] are socially negotiated as individuals, the community, and the broader culture interact (Engeström & Sannino, 2010). For instance, in the context of environmental advocacy, people may all be working to promote a particular policy to curb local diesel emissions (referred to as "the partially shared object"), but may understand what they are doing quite differently, or be working at the same time towards other goals that may vary considerably. One person might be participating in the activity to complete community service hours, another might have a child with chronic asthma, and another might simply enjoy being in the company of their friends, who happen to be working on environmental advocacy. These individual-level goals thus depend on an individual's experiences, values, and circumstances; as well as the surrounding enterprise in which the individual is embedded.

2.2.3 Culture and History

Culture and history are also important ideas in situated accounts of learning. How we act, what we take for granted, and what we see as important are filtered through the lens of culture – which is both inherited from our predecessors and also slowly shaped through collective activity. In the environmental advocacy example above, examining advocacy using either social practice theory or CHAT must also consider the broader cultural forces, which partially determine the mechanisms through which policies are made and challenged; or how a particular community is subject to high levels of diesel emissions in the first place. Sometimes goals are forced upon you, based on the social and cultural circumstances that you inherit.

---

[14] In social practice theory, goals are also socially negotiated and dependent on the broader contexts in which they are situated.

Most activities and practices (with the exception, perhaps, of biological ones) do not make sense outside of the cultural context in which they are situated. For example, the goal of "getting an A on an assignment" only makes sense when you take into account how achieving this goal might have cultural currency – perhaps a child's parent or caregiver will reward them; or perhaps over time, the child will be labeled as "smart." Moreover, this idea of "being smart" may imbue the child with beliefs about what she is capable of, which may (partially) determine the kinds of opportunities and life chances she will be able to access. Moreover, the very definition of "being smart" is culturally determined: if this same child lived within a nomadic, hunter/gatherer culture, grades would have no meaning at all, and very different markers of intelligence would be pursued. Culture does not only impact the goals of an activity, but also the norms/rules that people are asked to follow to achieve these goals, the roles they are able to take on, and the tools and artifacts they have access to.

2.2.4 Tools and Artifacts

Tools and artifacts are also important in both theories. Wenger (1998) describes artifacts in terms of the ways in which they reify and standardized work practice (*reification*).[15] As such, practitioners also *participate* in making the artifacts function in a variety of different scenarios. Similarly, within CHAT, artifacts, tools, processes, and so forth are also conceptualized as already existing with particular meanings attached to them, inherited from the broader culture (Cole, 1996), but the term *mediation* is used to characterize how they influence human activity. Regardless of the terms used (*reification/participation* versus *mediation*), both theories see tools as carriers of culture, shaping how a person is able to think and to act in a context. Tools – whether software, hardware, pedagogies, curriculum – all have ideas, and sometimes political bias (Winner, 1980), embedded within them.[16] At the same time, artifacts do not cause things to happen in any deterministic sense. Rather, artifacts *shape* the ways in which people engage in activity.

2.2.5 Political Dimensions

Finally, politics are an important part of context, to the extent that they determine who gets to call the shots and make the decisions that matter most. Politics can manifest in many different ways within social systems: they can shape the kinds of goals and practices that are

[15] Wenger argues that these artifacts may only be partially understood by the practitioner. For instance, Wenger gives the example of a medical claims form. While the person who authored the form may understand the intent of the categories and classifications, a claims processor may not, and yet the claims processor is still constantly making classification judgements to meet their quotas.

[16] One famous example of this, as described by Winner (1980), is Robert Moses's decision to build low-hanging overpasses, which effectively excluded busses from driving on the parkways he designed. This had significant equity implications, as people who used busses as their primary means of transportation – working class people – were therefore not be able to access particular beaches on Long Island (pp. 123-124).

considered valid (Gutierrez, Rymes, & Larson, 1995). They can influence whose ideas are taken up and whose are ignored (Langer-Osuna, 2011). They can impact what "being smart" means and the kinds of people who tend to be indexed as smart. They can influence how credit is assigned, how raises and promotions are determined, and more generally how resources are allocated.

Within situated accounts of learning, politics from the outside world – in the broader culture – are always present in learning contexts, and partially determine what is worth knowing and how it should be measured. This can have serious implications for equity: because people who occupy more dominant social positions tend to have more influence in making the rules, it is their cultural practices and assumptions that become normative within curriculum, software, language, social arrangements, and so forth (Gutiérrez & Rogoff, 2003; Gutierrez et al., 1995) (discussed in more detail in chapter 3). Moreover, there are also assumptions regarding how people are able to benefit from an educational program, which do not account for variations in relative power among students, or differences in access to opportunity.

There are various perspectives on how educational initiatives might account for and address the political dimensions of context. One approach, known as *critical pedagogy*, contends that a robust education must directly question and challenge the inequities present in the broader culture, versus treating them as natural and inevitable. Paulo Friere, a Brazilian philosopher and education reformer, is perhaps the most famous advocate of this approach. Freire argued that the purpose of education was to first learn to "read the world" – to see it as it really is – so that learners might ultimately "rewrite it" to be more equitable and just. In Freire's view, literacy is the means through which people could critically re-interpret their lives and experiences in relation to the broader culture, thereby developing a critical consciousness (conscientização). Friere contrasted this view of education to the status quo, which he calls the "Banking Model"– an indoctrinating form of education where students are treated as empty vessels in which to deposit facts (Freire, 1996). Freire also argued that the burden of liberating society – both the oppressed and their oppressors – fell on oppressed people, who were able to see the oppressive structures that the oppressors were blind to. Many initiatives have taken a critical pedagogy approach to education, where ideas rooted in language arts (Morrell, 2002), history (Gutiérrez & Vossoughi, 2010), mathematics (Gutstein, 2003; Rubel, Lim, Hall-Wieckert, & Sullivan, 2016), science, and computing education (C. H. Lee, 2012; Tissenbaum et al., 2019; Van Wart, Vakil, & Parikh, 2014) have been brought to bear on social justice issues relevant to the lives of students from non-dominant social groups.

Another perspective on the political dimensions of education involves expanding what counts as valid practice within a disciplinary context. For instance, within mathematics education, González, Andrade, Civil, & Moll (2001), question the curricular choices that privilege certain types of knowledge over others in the standard mathematics curriculum. These scholars argue that by drawing from students' cultural Funds of Knowledge, a broader range of mathematical competencies can be valued in the classroom, which will in turn make mathematics more relevant and meaningful to minority students. As we will see in chapter 3, Gutiérrez et al. (1995) make a similar argument regarding what constitutes a "current event" in social studies class. Within computing education, scholars have tried to better incorporate the cultural practices of indigenous heritage practices or practices that are typically attributed to women (e.g., K. A. Searle & Kafai, 2015).

Yet another political dimension involves disrupting power asymmetries within learning spaces, which is known as *relational equity* (Boaler, 2008). Relational equity emphasizes the importance of supporting students to develop equitable relations with one another based on mutual respect, trust, and accountability. Boaler advocates for a classroom structure that focuses on complex instruction, a teaching method that aims to disrupt broader gender and racial hierarchies from the inside out through the promotion of equitable classroom relations.

2.3 Methodology

In order to explore some approaches to fostering sociotechnical literacy development, I pursued an evolving set of questions and hypotheses intended to help young people examine and participate in a few different sociotechnical systems. Because one of my goals was to create a learning context that did not yet exist (as I envisioned it at the time), one methodology that I used was the design experiment. Design experiments within learning research are a way to "bringing about new forms of learning in order to study them" (Cobb, Confrey, diSessa, Lehrer, & Schauble, 2003, p. 10). According to Cobb, Confrey, diSessa, Lehrer, & Schauble – who are each learning scientists who do interventionist work – design experiments are characterized by (a) creating one or more conjectures about the process of learning, (b) designing the means by which that learning might be supported, (c) iteratively refining conjectures and testing new ones, and (d) ultimately, if all goes well, providing practical guidance regarding how a generative learning process might be instantiated (Cobb et al., 2003). For me, doing these design experiments involved designing and building software, creating learning activities, delivering these learning activities, and then studying the context in order to explore my various conjectures.

However, some aspects of my methodology were more ethnographic. Specifically, as I carried out the design experiments, I also engaged in *participant observation*, where I not only observed students, teachers, and community organizers, but also actively participated in teaching, and provided students and community partners with assistance. I draw from Erickson's (2006) idea of "studying side by side," to account for my active involvement, recognizing that while I did not have as much distance as a more "hands off" critical ethnographer might, my active involvement helped me to listen and respond to students' curiosities and interests. It also allowed me viscerally experience student excitement when it happened, as well as the tensions that emerged when the activities, technologies, and lessons that I introduced bored or irritated students. Spending time in the field with students, teachers, and local organizations also helped me to identify unproductive or off-based conjectures and refine (or abandon) them in future design experiment iterations.

I documented my field experiences by taking field notes; gathering student-generated artifacts; taking photos; and sometimes video recording students – while they were working, or while they were presenting their work. I sometimes recorded students' screens as they were working with various technologies, in order to capture their process. For two of the studies, I also interviewed students.

2.3.1 Reflections on My Biases and Positionality

In the qualitative, interpretive research tradition, it is important for the researcher to reflect on their their biases, assumptions, and positionality and how these might have influenced the research (Richardson & St. Pierre, 2005). For me, two sets of experiences in particular were deeply influential to this research. The first experience involves my journey to become a software developer, and the opportunities it opened up for me. After graduating from college, I knew almost nothing about computing, and found it inaccessible, intimidating, and uninteresting. It was only while working as a temp after college that I began developing some basic computing skills after being asked to maintain the office website. I found the work to be interesting and not as difficult as I thought, so I decided to enroll in a county-sponsored computing course while working part-time. There, amongst retired folks, high school students, and people who were between jobs, I learned to program in an environment that was comfortable, welcoming, self-paced, and had the right amount of structure and flexibility for me to learn skills and pursue my own interests. After a year of vocational training, I managed to get hired as an entry-level computer programmer, where I was able to keep learning about software development from coworkers and mentors. Ultimately these skills opened up many opportunities for me. Moreover, if *I* could go from knowing nothing about computers to becoming a programmer in a year – with only the help of a vocational training course – I believed that *anyone* could do it. I should also disclose that I am a white, upper middle class, Yale-educated woman, which has given me many advantages that I did not fully recognize when I was formulating these theories of the world.

The second involves my time as a software developer building civic technologies for federal, state, and local agencies. In one project I spent three years working on a data-intensive system that helped public institutions coordinate and collectively uphold important environmental and human protections within complex development projects (freeways, bridges, etc.). In another, I built a data "portal" to help local municipalities interpret climate data for their state-mandated climate adaptation plans. I believed in the work that I was doing, and that public planning was more accountable to people and to the environment in part because of the systems that I helped to build and maintain. Moreover, this identity of "benevolent software developer" gave my daily work (i.e. a big part of my life) meaning. Imagining the ways in which the software I was building benefitted society became a way (albeit an unverified one) for me to feel good about the work I was doing. That said, I never got to see *how* the state agencies were using these systems to make these well-reasoned determinations. This of course was because these determinations never impacted me.

When I entered graduate school, my interest in computing for the public interest and CSforAll found an easy home within the broader tech community, and within the fields of human-computer interaction (HCI), information and communication technologies and development (ICTD), and computing education. Many people – non-profit organizations, researchers, educators, and policymakers – shared (at least some of) my beliefs in the power of computing and computing skills. Moreover, with the seemingly endless stream of rapidly diffusing, accessible innovations for ordinary people (e.g. smartphones, social media and peer-production platforms with millions, even billions of users, and rapidly expanding broadband), everything had the potential to become more participatory and democratic. Everyone seemed to be becoming a tech-savvy content producer, and these were still early days. Within this context, I was eager to explore how this next generation of technologies could be applied to solve problems, and to support others to learn about and realize the benefits of computing for themselves.

2.3.2 Timeline of the Research

Over the course of the research, my conjectures about learning about computing and data in real world sociotechnical contexts changed significantly. My early efforts began with a fairly utopian assumption about the intrinsic value of data and computing knowledge, and its potential to promote social equity – much like the *CSforAll* initiative. Broadening participation would expand access to high-opportunity careers to young people from non-dominant social groups, which would ultimately lead to a more democratic and balanced set of voices and ideas within the public sphere. With this as a starting point, my early conjectures (described in chapters 3 and 4) were oriented around helping young people to see how powerful data and computing could be – for communicating issues and ideas, making data-driven arguments, and ultimately effecting change. To examine these conjectures, I became involved in a series of local initiatives during my time in graduate school, which shaped the overall arc of this research project (outlined in Table 3). Towards the beginning, I focused on designing and building accessible civic technologies, in collaboration with several non-profit organizations, to support young people and neighborhood groups advocate for critical public infrastructure (analyzed in chapter 3). These experiences fundamentally changed the way I understood how public agencies function, and the circumstances that might compel them to act. Whereas I had come into these projects as a civic tech evangelist, during my time working with these different community groups, I saw time and time again how data-driven arguments could be easily ignored if they weren't tied to funding mandates, already-existing policies, or other forms of power.

When students' computer- and data-mediated civic advocacy campaigns did not make the kind of impact that I had envisioned, I formulated new conjectures around how computing and data could be brought to bear on civic advocacy and activism, and partnered with new organizations. In one pilot project, I co-taught a tenth grade high school sociology course, where I examined how various qualitative and quantitative research methodologies could foster data and civic literacies. I also explored, in two different after school programs, whether building apps to address local social issues could productively foster computing and civic literacies (Van Wart et al., 2014). These efforts had mixed success (see chapter 4), leading me to conclude that tight coupling of technology and advocacy – as a pedagogical approach to situating computing and data literacies – was questionable.

In parallel to these pilot studies, I also spent time with students in more informal contexts. I became a mentor at the local Boys and Girls Club. My mentee, Oba,[17,18] and I first connected because of his interest in computing. However, our weekly (and sometimes daily) mentoring meetings, which spanned four years, covered all sorts of terrain relating to school, work, and life more generally. I learned many lessons from Oba as I followed him on his journey from high school to adult school to community college to work. Perhaps more than anyone, Oba helped me to ground my techno-utopian imaginaries in the real world. For three years, I also helped to organize and run an after school program, *Code 510*, to support middle and high school students pursue their own, computing-related interests. This gave me the opportunity to see what young people wanted to learn/do if I didn't force them focus on social justice (see chapter 4). Through this experience, I learned that the students I met were curious about all sorts of computing-related contexts and applications: to design and build interesting artifacts, to access opportunities, to participate in various cultural communities (via social media), and to understand how the various systems in their lives worked.

Table 3: Phases of the research

	Description	Duration	Referenced
1.	**Data, ICTs, and civic advocacy** Supported three action research projects that involved young people in various local advocacy efforts.	Jan - May, 2010 Jun - Aug, 2011 Jun - Aug, 2012	Chapter 3
2.	**High school mentoring** Volunteered as a tutor and high school mentor as part of a student-sponsored math and science outreach program	Feb, 2012 - Apr, 2017	Chapter 8
3.	**Data, design, and social change** Co-taught a tenth grade high school sociology course that focused on data, design, and research methods.	Spring, 2014	Chapter 4
4.	**Code 510** Organized and ran an after school program where college mentors and young people worked together to explore ideas around technology, coding, media, and digital design	Aug, 2014 - Dec, 2016	Chapter 4
5.	**Summer coding internships** Designed and hosted two summer internships organized around networked computing (described in chapters 5-6)	Jun-Aug, 2014 Jun-Aug, 2015	Chapters 5-7

Given what I had learned from the research and experiences described above, I decided to conduct a final design experiment that aimed to blend the creativity and expressive aspects of

[17] All names are pseudonyms.

[18] I met Oba when he was 17 years old. He is African American, and a few years older than the students who are featured in Chapters 4-7.

Code 510 with various real-world concerns around computer-mediated platforms web (described in chapters 4-6). Taken whole, the trajectory of this research has gone from considering technology as a political tool for direct social action, to one that centered the more playful and interest-driven motivations of young people, to something in the middle: helping young people to examine the possibilities and risks of networked platforms through creative production and communication. In the chapters that follow, I describe these explorations, and what can be learned from them, in greater detail.

3 SOCIOTECHNICAL LITERACY AND THE PURSUIT OF CIVIC ADVOCACY

In this chapter, I describe the initial set of studies that I conducted, which conceptualized computing and data as a way to empower local community groups – who had not previously had access to data-intensive technologies – to more easily document and communicate local challenges and ideas for addressing them. Because data and computing are often described as a way of making authoritative, evidence-based claims that potentially carry more weight in official spaces, I imagined this context to be an exciting way for young people to learn about these technologies, and bring them to bear on issues that impacted their lives and communities. I viewed this approach as following the logic of youth participatory action research (YPAR),[19] although young people did not get to pick their own topics.

The focal sociotechnical system characterized in this chapter is the management of public resources at the local level – including air and water; schools, parks, and community centers; and transportation and housing. Managing public resources is technically, politically, and logistically complex, shaped by formal and informal procedures, policies, and laws that are often mediated by data-intensive practices (Elwood, 2006). It also involves negotiating divergent priorities across diverse political constituencies, amid unequal power relations. I believed that context, because it is data- and people-intensive and because it has such a profound impact on individual and community wellbeing, would offer a situated way for young people to learn about data, and to examine whether and how data-centric ways of knowing figure into broader civic efforts.

---

[19] Through the YPAR model, young people study social issues that have a direct impact on their lives in order to explore the root causes of these problems and ultimately formulate solutions (Cammarota & Fine, 2010). A key commitment in YPAR is that students choose their own topics to study, which often involves investigating a problem that the learner is actually experiencing. Thus rather than doing research on people, Y/PAR highlights the importance of doing research with the people who are most impacted by the problem. This aligns with a Freirean model of critical pedagogy – where 'the oppressed' study the conditions of oppression in order to ideally create a more just world (Freire, 1996). As Freire argued, while science is often used as a force to regulate and control, scientific methods can also be appropriated to interrogate the mechanisms of oppression. As such, YPAR employs research methods – e.g. surveys, interviews, photovoice, and focus groups; and more recently digital tools, such as sensors and data from third-party sources – to study the nature of problems and formulate solutions. Some of the many strengths of YPAR are its attention to structural / root causes of issues, working with people who are most impacted by the problems under consideration (and assuming that those are the best people to research and design for change), and focusing on literacy is a tool to both 'read and rewrite the world.'

My initial set of conjectures framed the goal of the research as one of expanding *access* – i.e. *how can data and data-centric computing software be made more accessible and more inclusive of qualitative data?* (Van Wart, Tsai, & Parikh, 2010). In other words, I took for granted that these technologies and methods would be empowering (i.e. effecting local change), and that the primary task was extend their capabilities to better support the visions and ideas of young people. Therefore, I conducted two iterations of a design experiment, which focused on software design as a means of empowerment.

However, what I learned and intend to show in this chapter, is that some of the utopian ideas about data and computing, which I inherited from pervasive *scripts*[20] about the power of information, were naïve, deterministic, and overestimated the power of information alone to shift what people do and how they act. As such, my conjectures shifted to an examination of how ideas about the power of data and computing technologies shaped the practice of data science within public resource management.

A growing body of research has found that situated approaches to data education within various sociotechnical systems can bring these scripts to the surface, creating both tensions and possibilities within data science education. On one hand, situating data science in familiar, disciplinary contexts can help learners connect their local and experiential knowledge to new disciplinary ideas, build technical, methodological and spatial reasoning skills, and develop a more critical perspective towards the places they inhabit (e.g., Elwood & Mitchell, 2013; Enyedy & Mukhopadhyay, 2007; Lanouette, Van Wart, & Parikh, 2016; T. M. Philip et al., 2013; Rubel et al., 2017; Taylor & Hall, 2013). On the other, a situated approach can also lead students to challenge some of the assumptions that surround data science – particularly when data-centric forms of participation do not account for students' existing knowledge, experiences, and positionalities. For instance, students may already be quite knowledgeable of phenomena, such as residential segregation, that data science activities are intended to illuminate (Enyedy & Mukhopadhyay, 2007). Students may also reject the idea of data-informed decision-making, citing decisions instead made on the grounds of existing authority and power. They may reject conclusions drawn from data that do not resonate with their experiences and values (Rubel et al., 2017). They may also find that the assumptions embedded in data-centered activities neither reflect their notion of their local community nor their vision of how they might want to participate in helping their community (T. Philip, Way, Garcia, Schuler-Brown, & Navarro, 2013). In short, while situating data science in familiar contexts can be a productive way to learn about data, it can just as easily present challenges to data-centric ways of knowing and doing, which are important to acknowledge and understand.

To consider the possibilities and limitations of data and computing, I consider two cases where young people participated in various data-intensive practices including participatory mapping (Chambers, 2006), participatory sensing (Burke et al., 2006), and community ethnography

[20] Common-sense descriptions of data that imbue values and assumptions of the world, elaborated in the Conceptual Framing, and also in the *CSforAll* section above.

(Calabrese Barton & Tan, 2019) – as a means to understand and advocate for important shared local public resources. My analysis of these cases is guided by the following (revised) research questions, in light of what I learned during the design experiments:

1. What were the prevalent scripts around data in each case and how did they emerge?
2. How did the negotiations that surrounded these scripts shape students' engagement with data science?

In the sections that follow, I expand upon the conceptual framework, and describe the varied study contexts, design activities, and supporting tools. I then examine the different scripts that surfaced during the two data science-focused projects and how emergent tensions were negotiated. I conclude by discussing implications, considering not only how data science is introduced to young people but also how student critiques can be productively integrated into shaping a more robust and critical data science.

3.1 Conceptual Framework

3.1.1 Scripts, Counterscripts and Third Spaces

To help us examine why scripts matter and how they might be productively navigated within young peoples' learning activities, I draw on Gutiérrez, Rymes, and Larson's (1995) notion of scripts, counterscripts, and third spaces. Gutiérrez et al. define *scripts* as familiar and comfortable ways of knowing and acting in the world, which vary across contexts, cultural communities, and power relations. They demonstrate this idea by examining an activity in a high school classroom, where the teacher (an actor with more positional authority) implicitly defines being "knowledgeable about the world" as someone who knows the *Los Angeles Times* headlines. The authors ask us to consider how this definition is so easily taken as a given, and why it so neatly intersects with the teacher's own daily practice. They argue that certain cultural practices that constitute "knowing about the world" have become dominant or "transcendent" – defined by people who are in a position to make these determinations (often without even noticing). With no way into the learning activity and no opportunities to negotiate what counts as knowledge (given the teacher's seemingly immutable script), students create their own counterscripts, using the teacher's words for jokes and side conversations.

However, these clashing scripts – which each have something to say about the world from a particular perspective – also have the potential to "refract" and bend one another into a "responsive/collaborative" script, or "*third space*" (Gutierrez et al., 1995, p. 465). When this happens, new opportunities open up for genuine dialog across multiple knowledges, expectations, and values, thereby shifting what counts as knowledge within a social practice. While this third space is neither stable nor sustainable – more of an ideal than a steady state – these momentary shifts in perspective have the power to change how people think, thereby shifting the transcendent script, and culture itself.

3.1.2 Social Valences of Data

Scripts are also examined in the science and technology studies (STS) literature. As boyd and Crawford (2012) note, data science is often described in terms of its potential to advance human understanding and inform action. Fiore-Gartland and Neff (2015) elaborate on this idea by deconstructing these normative data science scripts into smaller units of analysis, which they call the *social valences of data* – specific expectations and values that explain how and why people gather, interpret, and marshal data towards particular goals. The authors argue that these valences can be assembled in different ways to help data practitioners make sense of their work. In this chapter, I examine three of these "valences," which I found to be prevalent across both cases:

1. *Discovery*: the idea that new forms of data analysis, applied to bigger, more diverse datasets, can lead to important insights and discoveries.

2. *Actionability*: the idea that data-driven discovery can subsequently be used to improve efficiency, service delivery, and lives based on facts.

3. *Truthiness*: the authority ascribed to any idea backed by data (over other ways of knowing, and regardless of the validity of the analysis).

Within my cases, these three valences worked together to constitute a version of the normative data science *script*, namely that data science can help produce and discover knowledge that can in turn be used to inform principled decision-making and action. However, just as the teacher's "current events" script described above reflected a privileged vantage point, this normative script also imbues beliefs about the world. Specifically, while this data science script might hold true in certain situations, data-centric ways of knowing also have a long history of being used by more powerful actors to authorize, maintain, and amplify unequal power relations (versus ameliorating them) (see J. Scott, 1998, for an extended discussion). Moreover, even when "everyday people" do engage in data-driven knowledge production efforts towards their own goals (e.g., farmworkers working to ban pesticides that are making them sick), their findings and their lived experiences are often dismissed as "unscientific" or irrelevant (see Irwin, 1995, for a review). To summarize, scripts about data tend to be optimistic and ahistorical, and typically do not attend to the broader social, political and historical contexts that also underlie action and change. As such, these scripts, when invoked in the "real world," have the potential to reify deterministic beliefs about data, while at the same time dismissing people whose life experiences do not align with this depiction of the world.

3.1.3 Local Ground: Designing for a Third Space

In designing for this third space ideal, Gutiérrez and Jurow (2016) argue that the activities and goals of learning (means and ends) must both be reconfigured so that students can make meaningful connections across contexts and between "everyday" and "scientific" forms of knowledge (Gutiérrez & Jurow, 2016; L. S. Vygotsky, 1978). I argue that participatory mapping offers one avenue to integrate these knowledge forms, allowing communities to collectively take stock of the places they live (Chambers, 2006) and locate themselves within a broader social and historical narrative. By allowing people to represent the many perspectives, critiques, and visions that they have, in relation to a place with a partially shared history and culture, the goal is to document and engage multiple knowledges and perspectives through data, towards a collectively negotiated end (Chambers, 2006). Digital mapping technologies enable a different set of place-based knowledge production activities, such as detecting patterns, sampling environmental features, and broadly disseminating findings. By bringing participatory mapping, data, and geospatial technologies together, I conjectured that I could create a context through which "everyday" and "scientific" knowledges could be brought into dialog, by supporting key data science activities (e.g., constructing data, conducting analysis, making inferences), and bringing the resulting analyses and representations to bear on students' local neighborhoods and communities (Enyedy & Mukhopadhyay, 2007; Rubel et al., 2017; Taylor & Hall, 2013).

To support this vision, I designed and implemented *Local Ground*, a participatory digital mapping tool (Van Wart & Parikh, 2013) that aimed to support youth bringing their own knowledge and histories of their local neighborhoods and communities into conversation with new data scientific techniques and disciplinary ideas. Central to the *Local Ground* design was supporting youths' engagement in the full range of data science activities, including defining a data protocol, collecting and analyzing data and visualizing and sharing findings. *Local Ground* was also designed to support multiple "everyday" and "scientific" forms of data, including photos, audio recording notes and drawings as well as more traditional data types such as spreadsheets and quantitative sensor data (Van Wart & Parikh, 2013). Several research groups have used *Local Ground* across multiple youth action research projects in the context of larger instructional designs with their own pedagogical and curricular goals.

Table 4: Data collected for each project group

Case	Duration	Data Analyzed
Air Surveyors	5 weeks (5 days/week)	Field notes Students' photos and notes Screenshots of data visualizations using *Local Ground* Transcript of final presentation Instructional materials and curriculum
Park Planners	16 weeks (2 days/week for 1 hour)	Field notes Participants' posters, photos, notes using *Local Ground* Transcripts of 3 final presentations

3.2 Methods

In this chapter involves a case study analysis (Yin, 2014) of two projects involving youth's data science activities. This analysis came out of a larger research project that involved designing mapping- and data-related software technologies to support project-based learning experiences. I began this research with a belief that data science could be an important tool for community empowerment, and that helping local community-based organizations to learn about data was a critical part of this agenda (a version of the normative script of data science).

Data sources include my field notes (Lofland, Snow, Anderson, & Lofland, 2005), audio and video recordings of group activity, facilitators' digital and written instructional artifacts, and student-created artifacts, such as their field notes, drawings, photographs, charts, graphs, maps and final presentations (see Table 4).

3.2.1 The Air Surveyors Project

In my first project, I worked with a with a five-week summer science program, which met daily for approximately three hours, to support high school students as they conducted an air quality study of the regional transit system, using particulate matter sensor data (Lanouette et al., 2016). The program's mission was to design community-based science experiences for middle- and high-school youth living in underserved communities that could help them to develop an understanding of important scientific concepts, and an appreciation of how scientific research contributes to addressing issues relevant to students' lives. The cohort I studied included 11 students between the ages of 15-18 years old, self-identifying as follows: Latino, Chicana, Mexican (7), African American (3) and Filipina (1) (see Visintainer, 2017 for additional analysis of this project). Students attended one of three large public urban high schools in the region, where 75-80% were eligible for free or reduced lunch. Several students enrolled in the program to make up science credits needed for high school graduation requirements. The lead instructor, Mark, self-identified as a white male, had designed and taught environmental science programs for several years.

3.2.2 The Park Planners Project

In my second project, I worked with a workshop-based city planning course, taken by undergraduate and graduate students, at a large public university. The goal of the course was to engage high school and university students in a "real world" city revitalization effort, which encompassed a public housing development, the nearby park, school, and community center. Serving as mentors, university students (myself included) worked with two 11th grade social studies classes, twice a week for 16 weeks, during their regularly scheduled class time. Approximately 60% of the high school students were identified as Latinx, 30% as Black or African American, 8% as Asian / Pacific Islander, and 1% as White / other. 80% of students were eligible for free and reduced lunch. The university students were a mixture of fifteen undergraduate and three graduate students, who self-identified as White (10), Latinx (4), and Asian (4).

3.2.3 Data Analysis

To analyze my data, I carried out two cycles of iterative coding (Saldaña, 2016), with the help of an undergraduate research assistant, using MaxQDA – an audio and video coding software. In my initial round of coding, I attended primarily to data science activities across the different phases of each project: (a) framing questions and opening activities, (b) gathering and analyzing data, and (c) sharing findings and making recommendations. I began my coding by attending to how each project was initially framed and how data was positioned within each project, using my field notes and the curriculum documents. Next, I examined students' data collection and protocol design activities (e.g., developing categories, keen observation), which I derived from field notes and from students' own data and notes. Next, I coded data pertaining to students' data analysis, including student-generated artifacts (posters, notes, collages, models), transcripts of in-class presentations and discussions, students' written reflections, intermediate charts and graphs, and my field notes. Across these multiple forms, I noted students' different types of data analysis (e.g., filtering, zooming, map making). Finally, I coded students' final project presentation transcripts, attending to how students communicated and marshaled data to garner support for their arguments and priorities.

Table 5: Air Surveyors' scripts and data science practices

	Example Scripts	Example Counterscripts	Data Science Practices
Phase 1: Introducing the project	Facilitator(s): "Science is a process of gathering and analyzing data to understand human and environmental health." **[D]** Community Partner(s): "Data can compel decision-makers to listen **[T]** and to act." **[A]**		Piloting the sensors, doing sampling, visualizing results, brainstorming causal relationships
Phase 2: Doing Fieldwork	Facilitator(s): "Look out for potential causes of particulate matter." **[D]**		Designing and adjusting the sampling strategy, sampling, taking notes/photos
Phase 3: Data Analysis	Facilitator(s): "Making charts and calculating summary statistics is a way to formally present scientific findings." **[T]**		Filtering data, authoring data visualizations (histograms, scatterplots, and heat maps), calculating summary statistics, researching explanatory relationships
Phase 4: Presenting	Decision-makers: Your findings are not scientific **[T]** Student: "You cannot make a change **[A]** if you don't know there's a problem." **[D]**	Student: "Keep buggin 'em 'til they change it." **[A]**	Pamphlets, presentations, scientific posters

The valences are coded as follows: T = Truthiness, D = Discovery, A = Actionability

From this initial analysis, it became clear that the scripts that surrounded each of the projects, and students' responses to them (including students' counterscripts) were fundamentally shaping students' experiences with data. Therefore, I did a second round of coding that attended to these scripts – noting whether any normative valences (Fiore-Gartland & Neff, 2015) showed up within them (thus connecting them to broader data science discourses), and any counterscripts that emerged (given actors varied values, experiences, and goals). My research assistant and I coded in parallel, and later discussed codes and themes together until they reached agreement. Examples of these codes – where scripts, counterscripts, and data practices are grouped into three project phases – are presented in

Table 5 (Air Surveyors) and

Table 6 (Planners).

Table 6: Park Planners' scripts and data practices

	Examples of Scripts	Examples of Counterscripts	Data Practices
Phase 1: Introducing the project	Facilitator(s): "Civic participation involves doing research [T] and working across constituencies to effect change." [A] Student: "We're young, energetic, know the area, and have fresh ideas." [A]	Student(s): "Why do we have to work on a random park?' Student(s): "They're not going to listen to us [T] or use my ideas" [A]	Students writing about their own experiences with public space (personal experience as data)
Phase 2: Doing fieldwork	Facilitator(s): "Data helps you to understand the opportunities and challenges [D], and to communicate them to others." [T]	Student(s): "I already know about this park. Why are we here?" [D]	Taking notes (drawing on maps, written notes), taking photos, discussing observations and what they meant
Phase 3: Data Analysis	Student: "Our park will create jobs, it will create artistic diversity, it will be all of these things all wrapped up into one little park;" [A]	Student: "I need a welcoming gate so people won't be like...that's only for white people." [A] Student: "I need places to feel free...and protected." [A]	Content analysis (via interpretive posters), class discussions of neighborhood investment vis-à-vis resource constraints; constructing new visions (informed by data and experience)
Phase 4: Presenting	Decision-makers: "I want to hear your ideas" [C] Decision-makers: "I want to know that you went through a process" [T]	Student: "Man, who's cutting the grass?" [A]	Data (mostly photos) only used when arguing for a particular public investment (benches, trees, community programs, etc.)

The valences are coded as follows: T = Truthiness, D = Discovery, A = Actionability

3.3 Findings

To show some of the ways in which the project scripts and data practices were emergent and mutually constitutive, I have organized my findings chronologically by case. Within each case, I focus on three phases: (a) framing questions and activities – where each project was introduced to students, (b) gathering and analyzing data – where the bulk of the data science activities happened, and (c) sharing findings and making recommendations – where students reflected on their findings and what they meant. As a convention, I have bolded the focal scripts and counterscripts, some of which are paraphrased based on field notes and curriculum documents, and annotated the valences within these scripts (e.g., *discovery*, *actionability*, *truthiness*) in italics. For simplicity, I use the term "script" instead of "valence" for the remainder of the chapter when referring to particular scripts.

3.3.1 Air Quality Project

I begin by examining the *Air Quality* project, which followed a fairly "traditional," quantitative, data science process, but which nonetheless revealed some of the scripts and counter scripts that might surround knowledge production efforts that are directed towards effecting change.

Framing questions and activities

The *Air Quality* project was initially conceptualized as a way to give students from underserved Bay Area communities the opportunity to participate in rigorous, hands-on science experiences that could be brought to bear on their lives and communities. To do this, the sponsoring organization partnered with a local environmental justice organization made up of residents and volunteers living in West Oakland – a predominantly working class, African American neighborhood adjacent to the Port of Oakland. Volunteers from the organization met with students during the first week of the project and described their neighborhood health advocacy initiatives and the role of data within them, including a successful effort to pass a city ordinance that banned diesel-burning semi-trucks from idling in residential neighborhoods. During these meetings, they explained that **data can be an activist tool for effecting change**, and invited students to join their ongoing air quality monitoring efforts, lending them several scientific-grade air quality sensors to conduct a study of their own. Through this process, the organizational volunteers established a pervasive script within the project that positioned data-driven knowledge production as a way to lend authority (*truthiness*) to civic advocacy (*actionability*).

Mark (the facilitator) also introduced an additional script during the first week, which positioned data science as a tool for *discovery* that could have significant consequences for community health and well-being. As he taught students to use the sensors, he invoked this script by saying (paraphrasing): **"Any of you can make a scientific discovery with these sensors. You never know what you'll find, or when you'll find it."** Mark subsequently challenged students to see what they might discover for themselves by walking around the neighborhood with the sensors. Taken together, these mutually reinforcing scripts framed the purpose of the data science activities and the potential contributions each student could make.

The *discovery* script appeared to resonate with students from the beginning, leading to inventive sampling methodologies and attention to extreme values. For example, during their first data collection activity, where students first piloted the sensors, students went out of their way to direct the sensors towards car exhaust fumes and dust from a leaf blower to find out how various emissions sources impacted air quality (in contrast to just walking around the block as Mark had instructed). These experiments turned into a spontaneous game that continued throughout the project: Who could capture the most extreme measurement? Given their curiosity about the sensors due to the invisibility of air quality, the creativity of their sampling strategies, and their mysterious air quality spikes visible in the *Local Ground* visualizations (Figure 3), students seemed invested in the *discovery* script and what could be learned from the sensors.

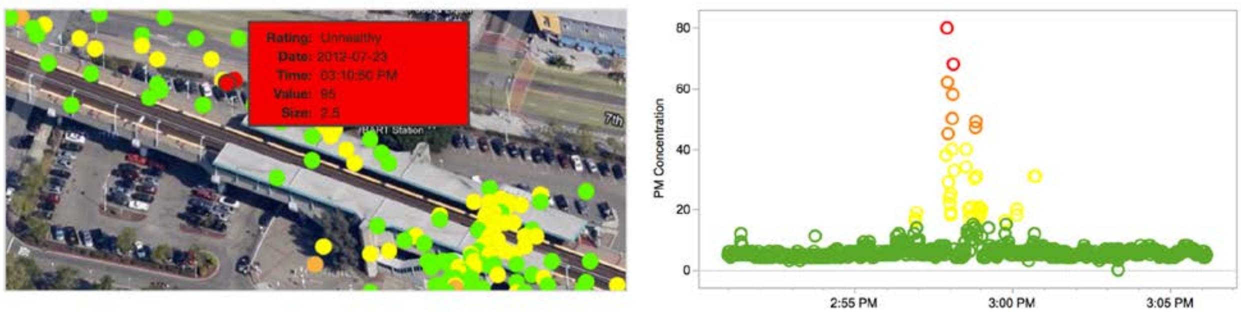


Figure 3: Map and scatter-plot view of particulate densities gathered by students during pilot sensor activity

Gathering and analyzing data

The *discovery* script continued to shape students' process of formalizing a sampling strategy, building directly from students' own curiosities (e.g., the impact of temperature, cloud cover, underground versus above ground, proximity to freeways, trees, and water on air quality). Students split into teams and rode the subway trains to their respective destinations, sensors in hand. By the end of their very first day of sampling, they identified what would come to be their key finding – the air in some of the underground transit stations was noticeably unhealthier than anywhere else along the transit system (see Figure 2 below). Mark cautioned that in order for this discovery to be taken seriously, it was necessary to understand what was actually causing the anomaly, and to demonstrate that their discovery was a regular, systematic pattern (*truthiness*). The overarching script therefore shifted from one of **discovery** to one that emphasized **proof and verifiability** (*truthiness*).

Students' emphasis on proof and verification (*truthiness*) in turn motivated varied data science practices including designing and revising their data protocol, repeat-sampling, and seeking out and documenting potential causal relationships. For example, each team assigned themselves a designated note taker and photographer to document potential factors that might influence air quality. One student, Maya, meticulously documented her observations at 30-second intervals as she rode the train, noting multiple potentially relevant variables on her notepaper, including how stuffy each train car felt and the number of passengers per car.

Sharing findings and making recommendations

The *truthiness* script remained dominant as students worked to communicate the importance and regularity their findings to various stakeholders. To do this, students used *Local Ground* to transcribe and link their field notes (Figure 4) to particular measurements, so as to connect their sensor data with what was happening when the sample was being collected. With help from Mark and me, students also used *Local Ground* and *Tableau Public*[21] to create summary statistics, histograms, heat maps, and scatter plots (Figure 5) to be used in various presentations.

Figure 4: Students' geo-referenced, transcribed notes of their field observations

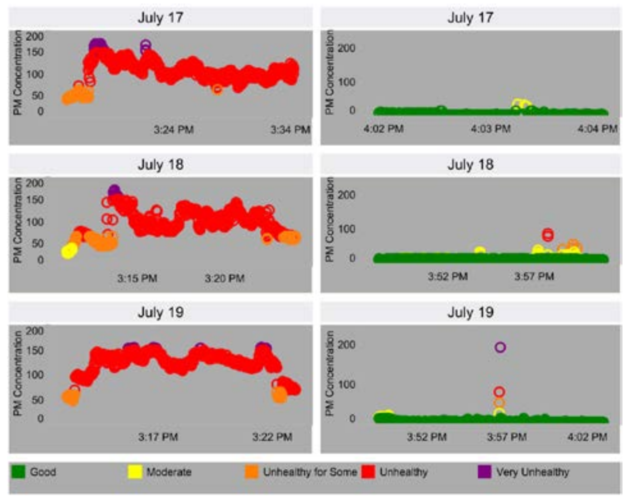


Figure 5: Scatterplot of two transit stations, showing a regular pattern of unhealthy air

As students shared their presentations with family and friends, the *actionability* script became salient again, namely the idea that **data can be a tool for effecting change.** However, this script was also challenged as students and audience members discussed whether and how their findings might lead to action and change. For example, as Leonard shared:

> The reason for us doing this project…is because we want to inform the people who have asthma and explain to them exactly what they're breathing in – because it can kill you and help them fix it by talking to the [transit authority].
> ...For me, personally, I want to follow up on this project by having someone come every few months or whatever, and take samples of the trains, and compare to other data that we had before…and you can actually tell if they listened to what you were saying. And if it decreased, you would know that

[21] https://public.tableau.com

> they changed something about the transit system, but if it increased, that means that you have to keep buggin' them until they change it.

Leonard, who figured the transit authority would probably have to be regularly monitored ("bugged") through continued data collection and analysis, therefore amended the *actionability* script with a caveat: **data can be a tool for effecting change – with persistence and other tactics.**

Following Leonard's presentation, another student, Melanie, used scatterplots (Figure 5) to show the regularity and extent of the discrepancy between above-ground and underground stations (truthiness). "As you can see," she said, "[the underground station] had more than 20 times the particulate matter concentration as [the above ground station]" (Figure 5). This tactic had an immediate impact, prompting an audience member, Temina, to challenge the *actionability* script:

> I don't know why we're measuring air quality with what the results you guys have come up with in your research, there's nothing being done about it! When I hear in the mornings, at 5 in the morning, that it's a "Spare the Air" day, I don't know what that means. Because it used to be that you didn't have to pay a fee to get on the [transit line]. And so, what good is it doing us to know!? And I think that's one of the questions that maybe tomorrow, or when you guys get to D.C., you can lobby and ask.

In Temina's view, public support for air quality protection was actually shrinking as knowledge of the issue continued to grow, building on Leonard's questioning remarks: **Can data be a tool for effecting change?** She also suggested that students take the transit authority to task by questioning them directly. Lelton (another student), responded to Temina, defending the actionability script:

> You cannot make a change, if you don't know there's a problem, so therefore you have to recognize it, and allow others to acknowledge it, so therefore the process of coming up with a solution can be even possible.

While agreeing that data-driven evidence did not inevitably lead to change, Lelton also argued that "acknowledging" problems and spreading awareness was not insignificant. Their study was creating the preconditions necessary to begin the process of solving larger social environmental challenges.

Following the presentation to friends and family, students put the ideas of *actionability* and *truthiness* to the test by presenting to regional transit and port authority staff. According to Mark (I did not attend this presentation), students' ideas were well received by workers at the Port of Oakland. However, staff at the transit agency (who were implicated in the findings) dismissed students' findings as "unscientific." That said, students found other outlets for their work that were more receptive, including authoring and presenting a poster at the American

Geophysical Union's (AGU) annual conference, and sharing of their findings with a national online news website.

Summary

In the Air Surveyor Project, an early discovery of an air quality hazard (*discovery*) pushed students to examine the regularity and extent of the issue (*truthiness*) and the factors that might be causing it. These pursuits motivated numerous data science practices: revising the study design, sampling, data analysis (sorting, filtering, visualizing, and calculating summary statistics), and eventually presenting to various scientific, municipal and community audiences. Through this process, students ultimately constructed their own idea of how data might support action should the transit agency ignore their findings (*actionability*), conceptualizing it as a way to help others understand the problem, or as a vehicle to hold the transit agency accountable, should their findings be ignored.

3.3.2 The Park Planning Project

The Park Planners' engagement with data was primarily qualitative. While qualitative inquiries are not typically considered "data science," some of the project scripts and counterscripts were surprisingly similar to those that surfaced for the *Air Surveyors*. As such, this case offers a useful point of comparison regarding how different forms of data and different ways of positioning the value of data might influence data science practice.

Framing questions and activities

The organizers of the *Park Planning* project worked with the City Manager's Office and the Housing Authority to craft a question that young people might be uniquely positioned to answer, given the city's ongoing projects and priorities. The question focused on an ongoing revitalization effort, and asked: How could the public housing development, nearby park, school, and community center be made to feel more connected to one another? To introduce the project, Shawna, an employee of the City Manager's Office and long-time Richmond resident, met with students, explaining that the city wanted them to participate in the revitalization effort and make recommendations the city could implement (*actionability*). University mentors (myself included) elaborated on this idea in subsequent in-class discussions, pointing out that the high school students had a wealth of local knowledge and ideas, as both young people and local residents, which were valuable but often overlooked within the planning process. Together, the mentors and the City Manager's office aimed to communicate that **students' ideas and perspectives mattered and could help guide the revitalization effort** *(actionability)*.

Some aspects of this script resonated with students, particularly the idea that their own knowledge and experiences were a valuable asset that gave them an important perspective on the community. As one student, Carla, expressed in an early written reflection: "We have young, fresh ideas. We know the neighborhood better. We have connections there. We know what we want so we know what we need in the space."

However, many students were deeply skeptical of the projects' goals and the *actionability* script that framed it, voicing several counterscripts that challenged the premise of the project. For instance, in a class discussion following the meeting with Shawna, students questioned whether anyone would care about, act on, or take their ideas and findings seriously. Students also criticized the relevance of the projects, questioning why *they* had to work on *these* problems, given the many other values and priorities they had. For example, one student wrote in an early reflection, "Why are we spending money on a random park? Why not my school?" (see Figure 6).

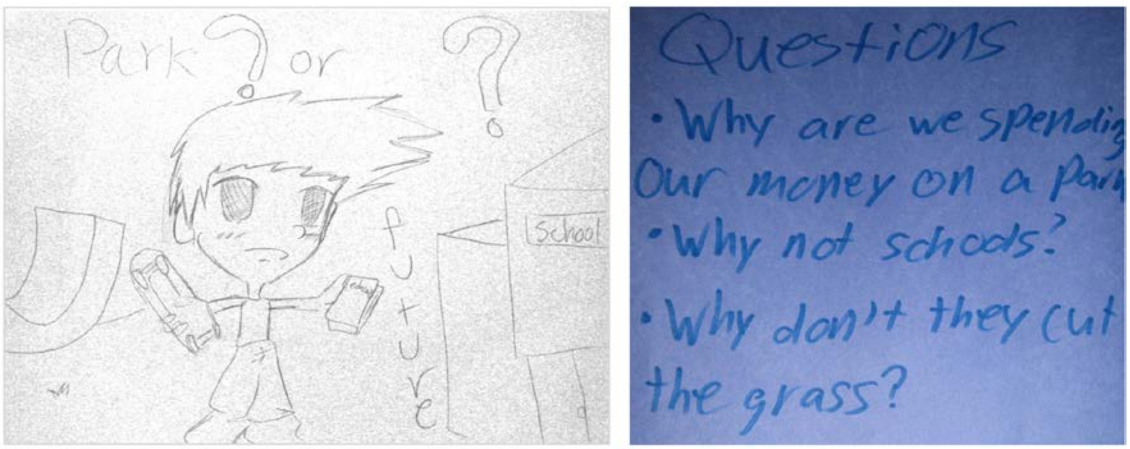


Figure 6: Students question the city's priorities, given their own priorities for their community

Gathering and analyzing data

During a subsequent field trip to the revitalization site, mentors introduced the role of data collection in the project, explaining to students that **data could be used to systematically document and communicate needs to others (truthiness), which was necessary to effect change** *(actionability)*. Mentors gave each student a paper map of the neighborhood and a worksheet to document the space, with prompts asking about which features of the environment made the space healthy and inviting as well as unhealthy and uninviting.

That said the site visit only gave students new material to critique the logic of the project. For instance, several students expressed that they found it "random" or "boring" (as written in note sheets) that they were being asked to walk around and document the park and surrounding neighborhood, asking: **Why are we here?!** Moreover, students were confused about what they were supposed to document, given that nothing was particularly noteworthy about the space from their perspective: **What do we even write down?** To them, the story was clear: they were standing in an unmaintained park. What was the value of systematically documenting it (*truthiness*) when there was nothing else new to say about it?

Mentors responded to students' counterscripts by asking them to elaborate on their criticisms and record them as field notes (i.e. data). They asked students: *What about this park makes it boring or random? What would you like to see here? What do you know about this place?* The mentors argued that there were specific, observable aspects of the space *made* the park and surrounding neighborhood felt a certain way, which were important to document in order to ground their subsequent designs and to justify particular design choices to decision-makers (truthiness). Thus, a kind of re-mediated (Gutiérrez & Jurow, 2016) version of truthiness script emerged, which encouraged students to document their criticisms (i.e. counterscripts) and interpretations, *and* their observations about the surrounding natural and built environment. Both perspectives were positioned as important, valid forms of data. As a result, most students eventually participated in the activity, documented their feelings about the park (e.g., that it was "boring" and "abandoned") alongside empirical notes (e.g., "the fences are ugly" and "there are signs of alert everywhere") (see Figure 7).

Figure 7: Planners document their qualitative data using Local Ground

The re-mediated *truthiness* script continued to guide students' data practices after returning from the field. During one activity, a poster-making session directed towards synthesizing and presenting their findings, students placed their ideas for the park and neighborhood (mostly drawings) alongside images (from their field notes and photographs) of things they wanted to change. For instance, one student wrote, next to a photograph of a partially intact chain linked fence: "I need places that have more trees and bushes that look nice. If we could have a place that we have to be free and not scared to go outside. And feel protected." Another student drew a picture of a sign saying: "Welcome to Richmond, City of Peace," which he pasted next to a photo of a sign, taken by a student, that read: "NO Loitering, drinking, begging, soliciting. Subject to fine" (Figure 8).

Thus, students' data analysis expanded to include additional ways of knowing – observations, personal experiences, feelings, perceptions, and so forth – to describe and symbolize their larger aspirations: feeling safe and free; being surrounded by beautiful and peaceful things; and having youth employment opportunities. Whereas data analysis had originally been positioned by mentors as a way to raise important issues to decision-makers, students directed their data analysis towards internal sense-making – to collectively articulate a (partially) shared set of concerns, experiences, and hopes for the city.

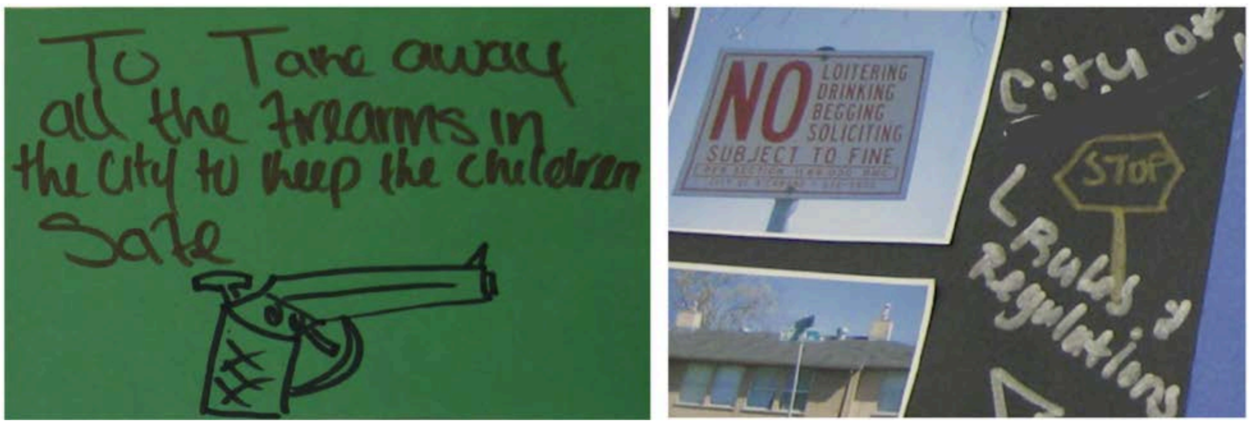


Figure 8: Sample images taken from posters created immediately following fieldwork

Sharing findings and recommendations

As students moved from analyzing the existing park and neighborhood to envisioning a future one, data science – broadly defined as a confluence of observations, experiences, memories, and hopes – made way for design. Students' counterscripts fell off, and students became noticeably more invested in the subsequent activities leading up to the presentation – taking on many different self-appointed roles (e.g., crafting designs, poems, speeches, digital- and paper-based data displays, and spoken word performances), and attending several optional design sessions where mentors met with students after school. Six different student teams prototyped their visions for the park and neighborhood by creating a series of physical models and maps. In presenting her team's model to the class, one student, Cindy, shared: "Our park will create jobs, it will create artistic diversity, it will be all of these things...all wrapped up into one little park." Mario, another teammate, added: "and we'll have play structures for little kids, a skate park, and a welcoming gate – so people won't be like... 'that's only for white people.'" Using *Local Ground*, these future visions were overlaid on an aerial image of the existing park (Figure 9), offering a way to re-frame the "random" space with their own beautiful vision of what it might become.

On the day of the presentation, a student, Jonah introduced the students' message at City Hall, to the City Manager's office and the Housing Authority:

> The presentations you are about to see are the product of hard work, dedication, and a true desire for change. We have investigated, we have mapped, we have analyzed, and we have discussed what [the neighborhood]

> looks like today and what we think it can become. We want change. We want to be heard. We are young, but we are RICHMOND–Real Intelligent Community Helpers Modeling my Neighborhood's Diversity. We are Richmond.

Following Jonah, over a dozen students presented – describing their respective visions for the neighborhood but using data only to the extent that it advanced a particular design recommendation. Through photographs and descriptions (i.e. data) of existing challenges (*truthiness*), students suggested that the city invest in community assets – a skate park, a community center, a "graffiti wall," and beautiful winding pathways and water features. They also asked for simple things, such as trashcans, someone to cut the grass, permanent bathrooms instead of porta-potties, streetlights, and moving the methadone clinic away from the children's playground. However, while the city officials praised the students for their energy, ideas, and dedication, when the park was revitalized a year later, many of students' most innovative ideas were not implemented.

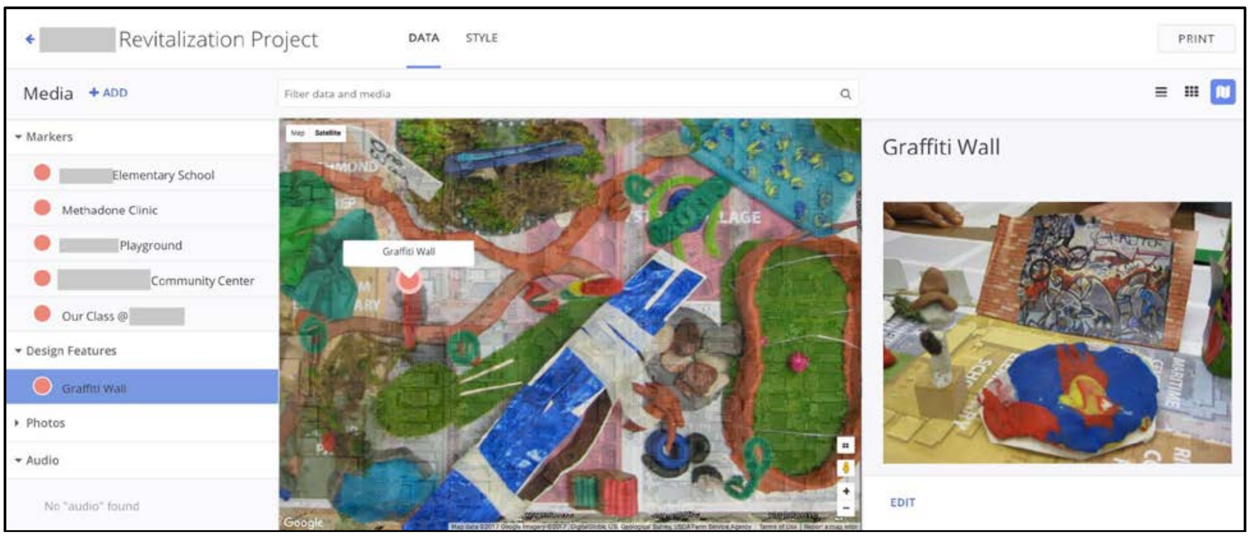


Figure 9: Local Ground allows images to be overlaid, annotated, and combined with other data. In this example, a representation of students' ideas for a park is overlaid on satellite imagery.

Summary

For the Park Planners, students' rejection of scripts that involved *discovery*, *actionability*, and *truthiness* led them to leverage the representational and communicative aspects of data to systematically examine a world they already knew, and to creatively explore and present alternative visions. Moreover, rather than doubling down on a cheery *make a change with data and design!* script, the facilitators (myself included) became more aware of and therefore open to reconfiguring the data activities to be more geared towards supporting student communication and representation of youths' perspectives. Through these tacit negotiations made possible through a structured-but-flexible participatory mapping process, students and facilitators were able to learn from each other: while students provided facilitators with a reality check, facilitators helped structure students' analysis and design process, and helped them to understand how their ideas were still relevant to the broader context of the neighborhood revitalization effort, thereby creating a *third space*.

3.4 Discussion

I have presented two situated accounts of data science learning, noting the rich data science activities that young people engaged in, and the negotiations that went into making the various data science scripts meaningful in light of three prominent data valences (*discovery*, *actionability*, and *truthiness*). Moreover, I argue that the reason these scripts surfaced and were negotiated in the first place was precisely because they were situated within real sociotechnical systems, where data science practice was considered in relation to the real work it was expected to do in the world. In this section, I revisit the efficacy of these scripts, and consider how they might be re-mediated to better account for students' lived realities amid larger structural forces.

3.4.1 Re-mediating the truthiness and actionability scripts

As we saw in the findings, the *actionability* and *truthiness* scripts were tightly interleaved in both cases: data was positioned as a way to justify and lend credibility to students' ideas and recommendations when presenting to public agencies. However, public agencies were *not* compelled to act on students' "data-driven" findings/recommendations in either case. For the *Planners*: while students presented compelling evidence that the park was not being adequately maintained by the city, and offered many wonderful suggestions for improving it, their favorite design ideas (e.g., a community "graffiti wall," an amphitheater, and a youth-run snack bar) were not carried forward (or even seriously considered).

It might be tempting to explain this outcome using the *truthiness* script: if students had only backed up their recommendations with more and better (quantitative) data, then perhaps they would have been more convincing to decision-makers. However, the *Air Quality* case challenges this script as well: despite engaging in repeated, quantitative sampling (involving the collection of over 100,000 air samples) and the production of a scientific poster that was accepted as *valid science* by the American Geophysical Union, the *Air Surveyors*' findings were still dismissed by the transit agency on the basis of not being sufficiently "rigorous."

These outcomes challenge a data science script that tightly couples data-driven knowledge production (*truthiness*) with change (*actionability*), absent larger shifts in power relations. Whereas students were able to control how their data were gathered, framed, and presented, they could not control whether their findings were viewed as authoritative by public agencies (*truthiness*), nor how local resources were allocated (*actionability*) – at least in the short term. This finding is important, given the presumed importance of data science as a privileged epistemic form (e.g., Irwin, 1995; T. M. Philip et al., 2013). These cases suggest that from a pedagogical perspective it is important to understand the limitations of data-driven argumentation, and to use this as an opportunity to consider the broader social, political and economic contexts within which these arguments are enacted.

These cases also provide us with tremendous insight into how data might play a role in meaningful action, where "action" is more broadly construed. In the case of the *Planners*, while students were initially skeptical of *any* change narrative, they eventually found meaning in their project by examining local injustices (through data collection and analysis), formulating their own visions for their neighborhoods and communities, and collectively working to voice ideas and concerns to decision-makers (a form of action). Along the same lines, the *Air Surveyors* eventually formulated a notion of meaningful action that entailed generating awareness (a precursor to change) through data and holding the transit agency accountable through regular monitoring. While convincing public agencies to act was not the sole purpose of students' efforts, I argue that students' interactions with the agencies motivated them to produce rigorous, compelling findings (*truthiness*); focus their message; formulate their overarching goals (*actionability*); and decide how they wanted to go about achieving them. One implication of these findings is that taking a first step towards a broader goal can also count as meaningful action. Moreover, determining what meaningful action means is something that can be discovered through tension and struggle, and can even be formulated in opposition to an initial idea that does not resonate.

3.4.2 Re-mediating the discovery script

Another prominent script in both cases was the idea of data-driven *discovery*: that through data science, new insights could be generated. For the *Air Surveyors*, this script aligned with students' experiences of the project. With the help of sensors and data analysis software (*Tableau* and *Local Ground*), students were able to discover a genuine air quality hazard, share this finding with local agencies and the AGU, and participate in every aspect of a very data-intensive knowledge production process. However in the *Park Planning* project, the discovery script needed significant re-mediation: students viewed the status of the revitalization site as self-evident and obvious. Whereas surprise and mystery undergirded the *Air Quality* project, many of the *Park Planners* perceived that they were being asked to discover things they already knew from their everyday experiences. However, when the discovery frame transitioned to one that involved taking stock of the site and connecting observations to broader themes in students' lives (a form of *truthiness*), it became meaningful.

3.5 Implications for Data Science Education

In this chapter, I have examined how young people engaged in data science, in the context of civic advocacy, with particular attention to the role of scripts and counterscripts in shaping data science practice. I argue that while some version of a normative script may always exist in data science activities, the resonance and relevance of this script depends on a number of situated factors. Providing avenues for students to critically examine these scripts, and reorient them (if needed), is an important aspect in the design of data science experiences for young people. While more research is needed to examine how these negotiations might be better supported, I highlight three aspects of my digital participatory mapping process that I believe were particularly generative:

3.5.1 Engaging in data collection

These findings illustrate that many of the most consequential (Gutiérrez & Jurow, 2016) data activities – including collectively analyzing the environment, thinking systematically about gathering information, articulating hypotheses, and even negotiating the very purpose of the project – happened as students gathered and analyzed data in the field. While data collection is often conceptualized as a rote activity that precedes the "real thinking", these case studies demonstrate that being in the field supported a collaborative, embodied way to develop a collective data protocol and a common analytic framework and hash out the larger goals of the projects.

3.5.2 Studying a context you already know

These findings also suggest that digital participatory mapping and sensing can be a productive way to bring students' own local knowledge and critical perspectives to bear on a number of key data science practices and concepts (Elwood & Mitchell, 2013; Rubel et al., 2017), and wrestle with some of the value propositions and contradictions inherent in data science (T. M. Philip et al., 2013) – namely the efficacy of evidence-based argumentation, and the idea of data-driven discovery. In doing so, we must remain cognizant that local, situated contexts are not simply sources of intuition to leverage for learning with and about data, but also complex, personal, and sometimes painful settings with long histories.

3.5.3 Reframing "discovery" as systematic inquiry

These cases also show us that the data-science-as-discovery script can certainly lead to a productive learning experience with data, as we saw with the *Air Surveyors*. However, systematically and collectively confirming ones' previously held knowledge and beliefs is also a powerful way to affirm one's own knowledge and perspectives, which might otherwise be dismissed (Irwin, 1995). While the Park Planners may not have discovered anything they didn't already know (Enyedy & Mukhopadhyay, 2007), collectively revisiting and analyzing the familiar helped them to reorient the project, develop confidence in their ideas, build solidarity, and formulate and deliver a collective message.

The task of situating data science often becomes one of finding generative intersections between domain-specific knowledge (e.g., science, mathematics, social science) and data-related methodologies, so that data science can be both and end and a means of learning. However, as critical data studies and learning scholars also remind us (e.g., Irwin, 1995; T. M. Philip et al., 2013), the value propositions inherent in data science do not always hold for people who occupy non-dominant subject positions, nor does the world solely operate on the basis of a well-reasoned argument. Therefore, providing opportunities for students and instructors to examine these scripts *within real sociotechnical systems* – through real-world practice that is open-ended and invites re-mediation (Gutiérrez & Jurow, 2016), and where each party has the agency to shape the nature and direction of the activities – can be a powerful way to make data science learning relevant and meaningful to a diverse cross-section of learners, with the potential of generating a third space.

4 REFLECTIONS ON "SITUATING" COMPUTING AND DATA LITERACIES

Chapter 3 examines a way of situating data and computing that aligns with a particular interpretation of critical pedagogy – where students examined and critiqued inequitable social arrangements (access to public resources) using new conceptual tools (data and computing) in order to advocate for their neighborhoods and communities. Thus, data and computing literacies were intended to serve as a means to help students cultivate a deeper understanding of their own lived reality, and as a way to unsettle experiences of injustice that were considered "normal." Through these projects, students were able to participate in various data-related and domain-specific practices and dispositions, develop a (partially) shared political identity (Elwood & Mitchell, 2013; Gutiérrez, 2008) as they crafted their message for decision-makers, and gain some perspective on the public process. However, these projects were not without their limitations.

4.1 Limitations

One challenge, particularly for the *Planning* project, was the fact that the direction and scope of students' project did not always resonate with students, in part because students were not able to determine the goal for themselves. Students certainly shaped how each project unfolded, but had also voiced many alternative priorities / goals they wanted to pursue (e.g. working to make their school better, working on community programs and businesses, etc.). Given that meaningful goals are a key driver of sustained engagement, precluding students from pursuing goals for which they had more control in determining seemed like a missed opportunity, and one for which many education scholars have long advocated (e.g., Dewey, 1938; Freire, 1996). Couldn't students learn about data and computing while pursuing ideas and issues that they were important to them?

Another structural challenge was the fact that the changes for which students advocated were not realized in either project, nor were there always clear avenues for students' to continue their advocacy work. For instance, in the *Planning* project, the program ended abruptly when the mentors' spring semester ended, which left some of the high school students feeling abandoned and frustrated. They had become very invested in their civic design ideas and in the collective solidarity they had built together, but then it was just over. While the project organizers and some students and mentors viewed the experience of "speaking truth to power" at City Hall as a reasonable endpoint, others viewed it as unsatisfying. In sum, effecting change at a city-wide scale was a high bar to clear, with numerous factors constraining the kinds of change that could be accomplished in the short-term. Could "taking action" be scoped in a way that could be sustained, and to better honored the constraints to which each program was subject?

Given these challenges, in the years that followed the *Local Ground* case studies, I explored whether there might be other ways of applying computing and data to sociotechnical contexts that had some bearing on students lives and communities, also using a design experiment methodology. The implicit question guiding these studies was:

How could computing be used for advocacy and collective action for young people?

At the time, I did not interrogate the premise of this question – namely that young people *wanted* or needed to engage in this particular application of computing, or that computing and data were somehow integral to advocacy (versus asking under what conditions it makes sense to engage in computer-mediated advocacy). Rather, I believed that any tensions that arose could be fixed by making adjustments to the tools, activities, and goals within a very particular configuration of computing education (versus examining the assumptions of the inquiry itself). In the process of actually doing these studies, some of my assumptions about the relationship between data, computing, and advocacy became more visible to me, and helped me to organize an after school program, *Code 510*, which aimed to be inclusive of *any* computing- and/or data-related goal, critical or otherwise (also described below). The lessons that I learned from these pilot studies formed the basis of the final design experiment, featured in chapters 4-7. In the sections below, I briefly summarize these pilot studies.

4.2 Tenth Grade Social Studies: Student-Directed Action Research

One of the pilot studies I participated in involved a revision to the way in which students' project goals were formulated. I wanted to know what would happen if students could examine issues that *they themselves selected* (versus having the issues pre-selected by the project facilitators). As Freire argued: "It's in making decisions that we learn to decide" (Freire, 1998, p. 97). Choosing project goals and topics was a way to foster greater agency and ownership within one's learning.

Figure 10: Selected student designs relating to transportation, mobility, and the environment

To explore this question, my collaborators (Tapan Parikh and Sepehr Vakil) and I co-designed and co-taught a tenth grade social studies class, titled *Data, Design, and Social Change*, which followed a youth participatory action research (YPAR) methodology (described in chapter 3, footnote 19). Over the course of a semester, students (1) selected a problem (e.g. disparities in educational opportunities, mass incarceration, racism, and other pertinent topics), (2) engaged in qualitative and quantitative methods (some of which were computer-mediated) to research the problem and identify its root causes (using surveys, interviews, quantitative data analysis, and so forth), and (3) formulated various designs intended to address these issues (e.g. public policy changes, increased public investment in education, the expansion of bike lanes, etc.) (Figure 10).

The course enabled students to pursue their own curiosities and interests, engage in a collective research process to make sense of their topic, and formulate a design idea to address a challenge in their school or communities. Data (broadly defined) provided a context for analytic discussions and sometimes made the regularity and pervasiveness of familiar inequities visible. Readings and discussions encouraged students to examine and debate their interpretations. Design pushed students to think about the underlying mechanisms of a problem, and which aspects of a problem students' solutions might address.

However, at the end of the course students ended up in a similar place: having a deeper understanding of the problem space (or else getting confirmation of what they already suspected) and a set of ideas for moving forward – presented via an interactive blog post of images, narrative, and data visualization – but no opportunities to connect with broader efforts or to actually instantiate their ideas. While formulating deeper understandings of issues they cared about was an important outcome for students, many again reflected that they were dissatisfied with the lack of action:

> "I didn't like the fact that I got very excited about the community and we don't actually get involved."
>
> "I wanted for us to do something outside like working with people in our community and do something big."

4.3 Apps for Social Justice: Computational Action

Following this YPAR project, I explore another set of approaches with my collaborators to examine whether and how computing and data might serve as a medium for *action*. Inspired by projects like *Hollaback!*[22] and *Stop and Frisk Watch*[23] – apps intended to raise awareness and advocate for changes in social norms and/or policy; and *Awaaz.De* – a mobile, voice-based system for smallholder farmers to share knowledge and learn from one another,[24] I thought

---

[22] *Hollaback!* – a platform that support women and LGBTQ+ people to share their experiences of street harassment (Dimond, Dye, LaRose, & Bruckman, 2013).

[23] *Stop and Frisk Watch* – an app created by the New York Civil Liberties Union to help people report being unlawfully searched by police.

[24] *Awaaz De* – a voice-based platform for peer-to-peer and expert-to-peer knowledge sharing (Patel, Chittamuru, Jain, Dave, & Parikh, 2010).

that by designing and building apps, students might be able to leverage data and networks to directly *solve* problems, thereby circumventing some of the institutional constraints[25] encountered in previous projects.

Furthermore, app development required an understanding of some important and fundamental computational- and data-related skills, which I believed were important to cultivate. Moreover, with the debut of *App Inventor* (Jeff Gray et al., 2012), a tool for authoring simple Android apps through a drag-and-drop interface, there were many stories circulating (many from the MIT Media Lab) about the potential of young people to design and build apps to improve safety, education, access to clean water, and so forth (e.g., Tissenbaum et al., 2019).[26]

Within this larger context of grassroots techno-optimism, I spent about two years working with different cohorts of high school students in various after school programs, helping them to design and build apps geared towards various social causes (e.g. Van Wart et al., 2014). This involved designing and building custom databases, and then using a variety of different "front end" tools (e.g. App Inventor, HTML/CSS/JavaScript) to implement the screens and interactions.

Though numerous iterations, I learned that app development certainly did help students to develop proficiency in various design, computing, and data-related practices, and encouraged students to think about the nature of problems and how they might be solved. Students also seemed to enjoy the process of app design and development, which included imagining how the app would help people, how it would work, and how it would look – including brainstorming colors, fonts, interactive elements, etc. (Figure 11).

Figure 11: Students explaining their initial, low-fidelity prototypes

[25] A prevalent idea at the time (e.g., Shirky, 2011) was that the emergence of mobile networked technologies and platforms would enable people to organize and act without the need for institutions or organizations.

[26] Notably, there was rarely mention of the people, institutions, pedagogy, ancillary help and support that was also needed to enable these social justice apps to come to fruition.

However, an apps-based approach to taking action did not solve the implementation challenges described in the previous studies. Rather, it just traded one set of issues for another. For instance, one consequence of designing apps to address social challenges was that students ended up separating problems into two categories: ones for which information might be the solution, and ones where it wasn't (access to resources, physical infrastructure, relationships, etc.). In essence, we asked students to consider whether and how issues they deeply cared about could be framed as information problems – which tended to preclude deeper examinations into whether requisite opportunities and supports were in place for people to act on this information.

We also learned that apps *did* in fact require institutional partnerships and organizational support. One student cohort, who worked for over a year to complete a working prototype of their college readiness and mentoring app (Figure 12), *Camino Contigo* (which translates to "Walk with You"), decided to eventually abandon their project after realizing the sheer amount of work needed to finish it, gather and maintain the needed data, and figure out how to partner with local schools / advocacy organizations. Moreover, students weren't even sure their app would actually help people, as that would require additional testing and evaluation (and more resources). Whereas the computational action (Tissenbaum et al., 2019) frame scarcely acknowledges the role of broader support networks in their account of taking meaningful action, the *Camino Contigo* project revealed that partnerships and organizational support were likely the most important part.

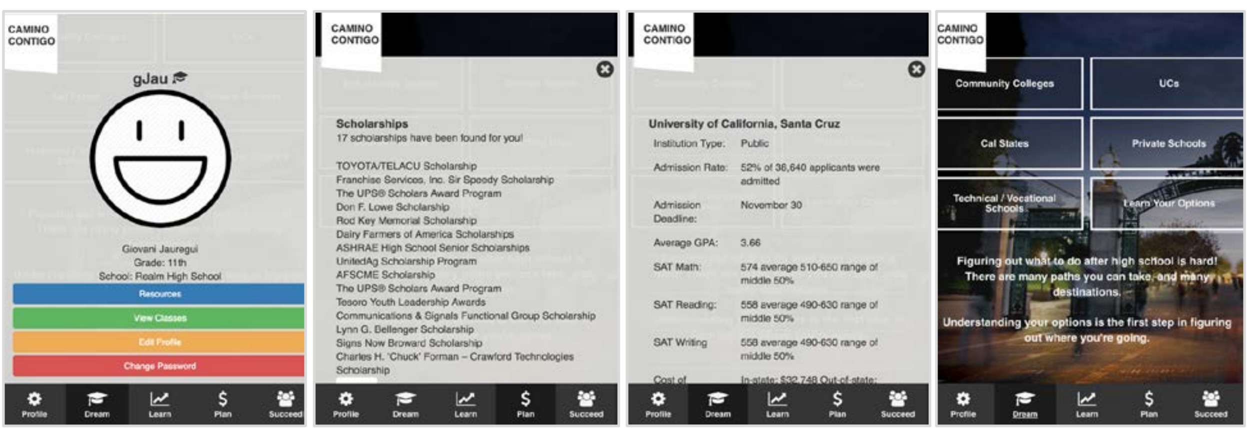


Figure 12: Students' high-fidelity prototype, built with HTML/CSS/JavaScript and a SQL database

Several members of the *Camino Contigo* team ultimately decided to pursue other, more accessible forms of community participation and action, which had more of a direct impact: mentoring middle school students, reinvigorating their school's Latino Alliance club, organizing protests, and hosting events that were deliberately designed to break down racial barriers at their school. In sum, the apps + social justice framing rested on two commitments that were not always compatible – one that privileged a technological solution and one that encouraged students to think broadly and holistically about designing for social impact. Moreover, "apps" needed institutions too, making them potentially just as complex (if not moreso) than other forms of social action.

4.4 Code 510: Student-Directed Creative Production

I co-founded an after school program for high school students, *Code 510,* which served as an institutional home for some of the pilot studies described above, including *Camino Contigo*. The program was run by graduate and undergraduate student mentors from various departments throughout UC Berkeley. My initial plan was to organize the activities around civic technologies, along the lines of some of the inquiries outlined previously. However, many the high school students resisted this framing almost immediately. They hoped that *Code 510* could be a place where they could pursue *their* needs, goals, and interests, which included hanging out with their friends, exploring various creative technical pursuits, getting a STEM-oriented extracurricular activity on their college resume, etc. As one student, Kira, explained to me as she negotiated her interests with our early vision for the program - she believed in social justice, but what she really wanted to learn about was how to make an RPG (role-playing game). *Was this the place for her and her friends to do it, or should they go somewhere else?*

Figure 13: Students present their work to family, friends, mentors, and faculty

In response to Kira's question/demand, the mentors, high school students and I negotiated and co-constructed a hybrid space: high school students could pursue any kind of creative technical or media project, so long as mentor could logistically support it. The project ideas were determined by students, while mentors helped with the practical and technical aspects (e.g. scoping, finding tools, frameworks, and tutorials, moral support, etc.).

Mentors also put their agendas into the mix by offering short, hands-on workshops throughout the year, based on their respective areas of expertise (e.g. tangible interfaces, eTextiles, data visualization, robotics, animation, web, visual design, etc.). By balancing short mini-lessons with self-directed projects, *Code 510* aimed to honor students' creative agency while at the same time introducing students to new ideas and teaching them some requisite skills/practices/ideas needed to actually complete their projects.

During the three years that *Code 510* ran, most students (with the exception of the *Camino Contigo* team) gravitated towards whimsical, playful projects (e.g. animations and games, and tangible user interfaces, and e-Textiles). For example, Kira's team built a virtual world that combined the team's shared interest in anime, art, and video games. The team spent six months creating original music, artwork, and coding JavaScript interactions to implement their interactive world. The next year, Kira and a few new students worked on a more sophisticated, 3D game using Maya[27] and Unity,[28] with the help of their mentor. At the end of each semester, *Code 510* shared their work with the broader community of parents, friends, and other students and faculty from Berkeley – to celebrate students and their accomplishments and to provide a deadline and goal that students could work towards (Figure 13).

Among the many lessons I took from this experience was how important it was to take students' (diverse) goals seriously. Meaningful goals (guided by the mentors) motivated students to engage with technically complex, difficult, and sometimes tedious tasks, often over the course of several years. Moreover, even just *having* a consistent, affirming space – where students could regularly experiment with various technologies and where mentors were invested in students' ideas – sustained participation over time, as students' priorities, curiosities, and knowledge evolved and grew. This model was also labor-intensive, with a 2:1 student-mentor ratio, and often required customized activities that mentors tailored to each student team. I wondered whether and how some of these lessons might be applied to more traditional learning spaces (i.e. classrooms), which were much more resource constrained.

4.5 Lessons Learned

Taken together, the *Local Ground* studies (chapter 3) and the pilot studies described above offer a window into some of the tensions and possibilities of data- and computing-related activities as they relate to critical inquiry, advocacy, and creative production. I have grouped these into two broad themes: alignment of means and ends; and goals and motivations.

---

[27] Maya (AutoDesk) – A professional modeling and animation software used in industry: https://www.autodesk.com/products/maya/overview

[28] Unity – A game engine used to make 3D games, VR, and AR experiences: https://unity.com/

4.5.1 Alignment of Means and Ends

By alignment of means and ends, I mean the extent to which data/computing-related skills (the means of participation) supported the goals (ends) of each project. Data collection and analysis were integral to the production of new knowledge for the *Air Surveyors*. However, for the *Planners* and the students in the *Data, Design, and Social Change* course, computing and data played only a supporting role – allowing important discussions and reflections to emerge, but only in conjunction with other ways of knowing (memory, experience, emotion, intuition, etc.). Regarding *taking action*, building an app was never as simple nor as effective as it was made to seem, and came with major logistical and institutional requirements that were not easily addressed in a youth-oriented educational program.

In sum, a tight coupling of technology and advocacy – as a pedagogical approach to situating computing and data literacies – was questionable. Why build a complex app when there were other more actionable ways to make a difference (e.g. community organizing, protesting, creating clubs and organizations, volunteering)? Why collect data when there were other ways of examining the world and students' lived realities (e.g. newspaper articles, historical essays, films, interviews with parents and community members)? Unless specific conditions were met (e.g. having logistical / institutional support, and a very specific question that quantitative data is well-positioned to solve), a social justice orientation led to numerous contradictions for both students and facilitators. When the means and ends aligned – for instance when students wanted to make something that computing/data was well positioned to facilitate (e.g. a game or animation), the *doing computing* aligned with the goal of *producing a computational artifact*. In this configuration, the artifact did not have to produce new knowledge or save the world – it just had to have some perceived value from the perspective of the maker; to be fun, interesting, or useful for its intended purpose.

4.5.2 Goals and Motivations

Another important takeaway from these studies was that honoring students goals could be a tremendous source of motivation that could inspire sustained engagement in any number of practices. Any number of goals could be inspiring to students, including directing their efforts towards a meaningful social justice issue, a creative endeavor, an aesthetic goal, or as a way to connect and communicate with other people. Goals could also be discovered (versus known a priori) as learners gained more insight into new ideas, problems, and topics. Goals could shift and change, shaped by students' interests, prior experiences, and by the relationships formed with other people and with the topics / problems under consideration. The inverse was also true: goals that were uninteresting and immutable would almost certainly lead to disengagement, lest they could be re-mediated and made relevant by students and facilitators (chapter 3). Therefore, activities were most generative when students had some agency and control over the goals, and adequate supports to pursue those goals.

5 SOCIOTECHNICAL LITERACY AND THE DEVELOPMENT OF "THIRD-PARTY" APPS

The empirical studies discussed so far suggest that my approach to sociotechnical literacy development, where computing and data were treated as a means to pursue social change, needed to be rethought. Perhaps there were other ways to foster criticality that better connected to the kinds of tasks which data and computing were well-suited (e.g. computation, communication, information dissemination, and creative production). One finding that came out of the *Data, Design, and Social Change* pilot work was that students had a knack for examining and critiquing design solutions – and even seemed to enjoy it. Therefore, I wondered if young people might find it valuable to take a closer look at existing computer- and data-mediated systems, in order to analyze the assumptions, goals, and outcomes of these systems. Many important social and critical issues could be examined from this perspective – fake news, privacy breaches, scams, monopolistic practices, amplifying the agency of bad actors, etc.

I also wanted to be mindful of students' interests and goals: whereas applications that leveraged *data* (i.e. platforms and distributed systems) would provide greater insight into the role that computing might play in organizing, communicating, influencing, etc. human activity, pursuing more playful applications (e.g. games, animations, and aesthetics) could be a generative source of engagement and self expression for many students. I wanted it to be possible for both kinds of participation to exist together.

To balance the affordances of each kind of computing activity – data-intensive computing and interface design and development – I built on an idea that was proposed by one of the *Code 510* students, Anna, who was then a high school junior. She had become interested in (and personally affected by) issues involving immigration rights and DACA (Deferred Action for Childhood Arrivals), and wanted to start a social media campaign to correct the absence of positive/everyday stories about Latinx immigrants, and to use her existing distribution channels (Instagram, Facebook, Twitter, etc.) to circulate these stories. She and her team also had some very specific design ideas for a website they wanted to build, to highlight and amplify all of these social media posts, which reflected their team's unique aesthetic tastes.

I realized that the genre of computer- and data-mediated participation that Anna was describing could accommodate many interests and types of applications (of which advocacy was one possibility), while at the same time allowing for a critical examination of networked platforms. I formulated the following conjectures:

- **Networked data and media platforms** would allow students to access a wide variety of different sources of familiar content and media (as I could conceivably help them to hook into the data of any content provider), which students could leverage and curate according to their interests. This would also expose them to web architecture and some data modeling ideas.

- **HTML, CSS, and JavaScript** would enable students to design interfaces for integrating and displaying this varied content, which could be very creative and expressive. This would also help them learn about coding and formal languages.

- **The resulting third-party apps** could serve as a generative site of critical reflection – where code, data, and social values could be considered together, including how each shaped the other.

To examine these high-level conjectures, I designed a learning environment that aimed to foster sociotechnical literacy development, with the intent of also honoring both the Annas and the Kiras of the world.

5.1 Learning Goals

To create a context for computing that could motivate an in-depth look at both the social and technical dimensions of platforms and networked systems, I designed a study to help high school students learn about two broad categories of ideas. First I wanted students to develop a set of computational skills, strategies, and dispositions intended to help them reason about existing systems and participate in building their own simple web apps. These included (1) familiarity with code, (2) familiarity with data, and (3) a high-level understanding of web architecture. Second, I wanted students to consider how some of the more familiar and ubiquitous applications of networked computing have become so successful, and their potential impacts and consequences. This included (4) exploring different applications of networked computing, and (5) considering some social implications of networked apps, including incentives, economics, privacy, and so forth.

5.1.1 Familiarity with code

Familiarity with some of the formal languages used in client-side web programming (Nardi, 1993; Dorn 2012) – specifically HTML and CSS – would make it possible for students to be able to inspect and experiment with various web concepts and techniques. Learning some programming ideas would be a useful way to make computing applications visible to inspection and experimentation. Through programming, students could concretely instantiate and "run" their computing-related ideas, while building artifacts that they could share and feel proud of (Papert, 1980). Moreover, by gaining some hands-on experience with code, I hoped that students would develop other useful practices – e.g. debugging, tinkering, and reading and understanding technical documentation – that they could carry forward into new computing contexts.

Rather than helping students to become programmers as typically conceptualized in an introductory, undergraduate CS1 course,[29] I was more interested in adapting learning research

[29] Many of the current computing education initiatives, even those geared toward children, derive from some version of a traditional CS1 programming course – the first course taken by undergraduates majoring in computer science to prepare them for more in-depth study of computer science. CS1

from the workplace and in informal learning settings. For instance, Bonnie Nardi, in her studies of programming in workplace settings (1993), found that many domain experts who were new to programming easily learned to use formal, domain-specific computing languages (DSLs) in the context of completing job-related tasks. Nardi (2003) refers to these kinds of programmer-learners as "end-user programmers" – people without formal CS training who nonetheless perform complex computational tasks in the context of their daily work. A characteristic of DSLs is that they provide useful abstractions to simplify relevant tasks (Van Deursen & Klint, 2002), and tend to be declarative – where users can simply express rules using higher level semantics. In practical terms, whereas a general-purpose (i.e. "Turing complete") programming language, such as C++, would require many lines of code to calculate a statistic like "average salary" (Figure 14), the same task could be accomplished much more simply with SQL – a DSL specifically designed for data manipulation tasks (Figure 15).

```cpp
#include <iostream>
#include <fstream>
#include <string>

using namespace std;

ifstream in_file;

int main() {
    char newline = '\n';
    char delimiter = ',';
    string row;
    string name;
    string year;
    int salary;
    int total_salary = 0;
    int count = 0;
    in_file.open("employees.csv", ios::in);
    while (in_file) {
        in_file >> row;
        name = row.substr(0, row.find(delimiter));
        row = row.erase(0, row.find(delimiter) + 1);
        salary = atoi(row.substr(0, row.find(delimiter)).c_str());
        row = row.erase(0, row.find(delimiter) + 1);
        year = row.substr(0, row.find(newline) - 1);
        if (year == "2018") {
            cout << name << ": " << salary << endl;
            total_salary += salary;
            ++count;
        }
    }
    cout << "Average salary: " << total_salary / count << endl;
}
```

Figure 14: A C++ sample for calculating average salary

```sql
SELECT avg(salary) FROM employees
WHERE year = '2018';
```

Figure 15: A SQL (DSL) sample for calculating average salary

---

covers the fundamental building-blocks of modern programming – I/O, expressions, control structures, functions, data types/structures, and recursion (Tew & Guzdial, 2010). These concepts are important prerequisites for more in-depth study of computer science, including writing production-quality software, designing new programming languages, building compiler systems, etc. However, a focus on the fine-grained details of programming also means that the exploration of higher-level systems and applications, including the ones that novice learners are accustomed to using in their everyday lives (apps, websites, etc.), are delayed. Moreover, this low level of abstraction is not necessary in order to understand higher-level tasks like data manipulation, building interfaces, and examining web architecture.

For end-user programmers, computing is often seen as a necessary skill, required to get a particular job done but not a pursuit in itself – a means to an end. Many of the students with whom I worked were interested in learning about how to create something very specific with code. As one student, Stella, put it: "I would rather write the story and have somebody else build it than the other way around."

5.1.2 Familiarity with data

Given that data is a key driver of networked platforms and web services, a second learning goal that I had for the design experiment was to help students learn some data-related ideas, including the purpose of data, how it can be represented (e.g. via JSON) and used – whether to represent an idea (see "data modeling,", chapter 1, p. 7) or to create secondary information products or apps (see "data-as-a-service," chapter 1, p. 9). This could be achieved by helping learners become familiar with the kinds of data made publicly available by commercial content providers, and understanding some of the ways in which these feeds are and can be used.

5.1.3 A high-level understanding of web architecture

A third learning goal that I wanted to support was to help students gain a basic understanding of web architecture. This included an awareness of the many different types of resources (media, data, and code) and entities (e.g. companies, individuals, news agencies, non-profits, etc.) that exist together in a single webpage, and how information can be displayed and/or collected through client-server(s) and server-server interactions. My hypothesis was that by helping students to think about the *mechanisms* through which bidirectional – and often invisible – communication happens between client and server, they would be better able to reason about the possibilities and vulnerabilities of their everyday web activities.

5.1.4 Applying technical to one's own interests and goals

Another goal was for students to see themselves and their values represented in the activities and goals of the course. This could encompass new ideas for systems that students might want to build, a greater interest in computing-related ideas and practices, or a different way of thinking about or participating in social media. Ideally, students would carry forward some of these skills and critical stances into future engagements with systems, as either consumers or producers of these technologies.

5.1.5 Considering the values, incentives, and economics of platforms

The business of social media platforms (e.g. Facebook, Instagram, YouTube, etc.) is to recruit as many users as possible, create data profiles of each user, aggregate these profiles into classes, and sell targeted ads (Gillespie, 2010). In this configuration, users "pay" for the service with their data. Alternatively, some commercial platforms (e.g. Spotify, Netflix) charge a fee to fund their data and information services, while others (e.g. the Rijksmuseum, data.gov, Wikipedia) are made freely available, funded by government or non-profit entities. I wanted young people to understand these different models, and consider the pros and cons of each. How might money and incentives intersect with what is technically possible to produce in the landscape of networked platforms?

5.2 Design Principles

To help connect the learning goals to students' more immediate interests and values, I followed three design principles:

1. Providing multiple avenues to engage with social and technical ideas
2. Working with familiar, "everyday" systems
3. Following a use-modify-create learning progression

These principles provided guidance for organizing and selecting the activities, tools, and strategies to be inclusive of students' diverse goals while making the social and technical ideas tractable and relevant.

5.2.1 Providing multiple avenues for engagement

Researchers who study broadening participation in computer science often cite the importance of "wide walls" (Papert, 1980) – to support "many different types of projects so people with many different interests and learning styles can all become engaged" (Resnick et al., 2009, p. 63). "Wide walls" make it more likely for someone without programming experience to see computing as a way to participate in an existing interest; or for someone who typically gravitates towards more technical practices to consider and reflect on the applications and implications of computing. From my pilot work, I found that some of the key dimensions that promoted "wide walls" were *creative expression*, *content*, and *communities*. Therefore, I tried to provide opportunities to engage with all three.

To support *creative expression*, I integrated many web design activities into the course to introduce students in different visual / interaction design techniques. Regarding *content*, I tried to select data and media platforms that offered a broad variety of content themes (e.g. news, gossip, art, music, and entertainment) and forms (e.g. videos, audio files, text, images) so that students might find something that interested them. Finally, I hoped that working with popular social media platforms would also provide opportunities for connection to relevant *communities* (Ito et al., 2010; Kafai & Burke, 2014) – whether that be their immediate social networks, or more aspirational ones, like online groups organized around shared interests or popular culture references.

5.2.2 Working with familiar, "everyday" systems

A big idea in sociocultural theory is that the development of "scientific" concepts involves making sense of concrete, everyday experiences in relation to a more systematic framework, typically introduced by a "more experienced other" (e.g. teacher, parent, etc.). As Vygotsky describes:

> In working its slow way upward, an everyday concept clears a path for the scientific concept and its downward development. It creates a series of structures necessary for the evolution of a concept's more primitive, elementary aspects, which give it body and vitality. Scientific concepts, in turn, supply structures for the up-ward development of the child's spontaneous concepts toward consciousness and deliberate use. (Lev S. Vygotsky, 1986, p. 194).

Thus, "everyday" concepts give scientific ideas "body and vitality" (or context), while scientific concepts provide new ways of understanding experience, and thinking about systematic relations. Put another way, learning is facilitated when both bottom-up (i.e. critical mass of experiences) and top-down (i.e. systematic knowledge, transmitted via language) concepts mutually inform one another. With this idea in mind, I hoped that working within a familiar, "everyday" systems would ground ideas involving data, computing, and web architecture (i.e. "the scientific') in the concrete (Lev S. Vygotsky, 1986). Studying the systems students already used (as well as ones they didn't) could allow them to map new social and technical terms onto the concepts they already knew, and to stretch themselves to consider some of the opportunities and issues associated with these systems. As Muibi, Dorn, and Park (2015) note:

> Students' lived experience with [computing] is one that is richly interactive, visually appealing and highly reliant on third party APIs, but the artifacts they are able to construct with just basic knowledge of [the underlying languages] do not reflect their view of the "real world" both in terms of the final product and in terms of the tools they know are widely available. Thus if the primary goal...[is] to ensure that students have a positive experience while introducing them to some basic computational principles and tools, it appears necessary to explore new tools that more readily strike a balance between student expectations and instructional goals of code-oriented learning objectives. Rather than fight to redefine student expectations...perhaps it is better to embrace those expectations as important motivators and find ways to bridge the gap (Muibi et al., 2015, p. 236).

5.2.3 Use-Modify-Create

Use-modify-create (I. Lee et al., 2011) is a pedagogy that blends two approaches to learning that each have significant strengths within computing education – *constructionism* and *direct instruction*. Constructionism is a form of learning by doing (i.e. making) that engages young people in the construction of objects and computational artifacts. The idea was developed by computer scientist Seymour Papert, perhaps the most well-known figure in computing education for children. Papert essentially extended constructivist learning theory (in the lineage of Piaget) by arguing that with the right computational medium (e.g. Logo, programmable bricks, etc.), it would be possible to "create the conditions for children to explore, naturally, domains of knowledge that used to require didactic teaching" (Papert, 1980, p. 187). Many computing education efforts (e.g. Scratch) have been based on constructionism, emphasizing the importance of building sharable, computational artifacts through self-directed learning.

However, self-directed learning has also been critiqued by both computing education and equity scholars. Within CS education, for instance, lab studies have shown that many ideas are best learned via direct instruction (Guzdial, 2018). Moreover, self-directed learning arrangements tend to advantage those who already have prior knowledge of the practice, or who have access to supports outside of the practice which are not available to everyone. As Lisa Delpit (2006) argues, because exploratory forms of learning (e.g. constructivist approaches) do not adequately support skill development, they disadvantage students who don't have the same opportunities to access these skills outside of the classroom – particularly if students' language and cultural practices at home differ from those valued in the classroom (an instantiation of "the culture of power"). On the other hand, as Friere argued (2.2.5), direct instruction can easily turn into "The Banking Model of Education" (Freire, 1996) and lose its "body and vitality" (Lev S. Vygotsky, 1986).

In my view, the idea of use-modify-create, proposed by education researcher Irene Lee and colleagues (2011), offers a solution to some of the challenges to constructionism and direct instruction named above – namely balancing learner agency and curiosity with explicit guidance and skill-building. Within this model, learners begin by first *using* an existing computational artifact, built by an expert, which allows them to understand what they will be building and how it should work. Next, students are asked to *modify* particular aspects of the artifact by changing parameters, adding or removing code, and so forth. During the modify phase, the pedagogy can direct students' attention towards particular features of the artifact, so that relevant skills and practices can be introduced in relation to the specific forms/functions they enable. Finally, once students have practiced a series of modifications, they are asked to *create* their own (similar) artifact. While the *use* and *modify* phases honor the importance of expert guidance and making practices and skills explicit (akin to apprenticeship), the *create* phase has many of the same commitments to learner agency as the constructionist approach.

In the context of the "real world," networked platforms and the APIs they make available provide an implicit framework for scaffolding a use-modify-create learning progression. Because students were already *using* these systems, a clear next step was to deconstruct these systems' innerworkings (e.g. data, code, and networks), examine how each aspect worked, and ultimately put the pieces back together again to create something new. To support this vision, I created a series of templates and code samples that would facilitate students' ability to create their own computational artifact. By progressively customizing existing templates and samples, I hoped that students would ultimately develop the skill, practices, and perspectives that were needed to create their own data-driven web apps.

5.3 Supporting Tools and Resources

I leveraged many different tools and activities to allow for progressive levels engagement with different social and technical ideas. Each abstracted aspects of web applications that were not immediately relevant, in an effort to help students to attend to one idea at a time. The overall goal was both to help students build their own web applications, and to examine existing ones, by providing background knowledge and experience working with the different pieces –

HTML, CSS, and JavaScript; authentication protocols (e.g. OAuth); REST APIs, binding data to DOM elements, etc. Below, I describe six of these tools and resources in more detail:

1. Cloud-based everything
2. Chrome's "Developer Tools"
3. Starter code and tutorials
4. Code snippet catalog
5. API Tutor
6. Commercial platforms

5.3.1 Cloud-based everything

Because I hoped that some of the activities explored in the design experiment could be used in classrooms – which in my experience typically used low-cost Chromebooks – everything was done in the cloud. File management and programming was done via a free, cloud-based editor called ShiftEdit.[30] Each student had their own server directory, from which they could create, edit, and delete files. Therefore, students' files were always available to them – at home and at the internship site, without the need for software installation or configuration. All of the helper tools used were also cloud-based, making things logistically easier and portable.

5.3.2 Chrome's "Developer Tools"

One of the most useful tools for studying *existing* platforms was Chrome's *Developer Tools*, which were built into the Chrome browser. Using the "Elements" panel of the developer tools, students were able to examine and modify the HTML and CSS code for any web page, by highlighting a DOM element and tinkering with it. For instance, in Figure 16, a student has modified the text of a button on the Google homepage from "I'm Feeling Lucky" to "I'm Feeling Hecka Tired."

The "Network" panel of the *Developer Tools* allowed students to see the sheer number of resources being loaded on any web page, and that many of the larger websites were sending data back to a variety of different servers. For example, in Figure 16, the network tab shows that the ESPN homepage made 462 different communication requests to send off data and to load web resources – images, videos, scripts, documents, stylesheets, etc. The network panel also allowed students to inspect the many third party applications and services that were present in a single web pages.

---

[30] https://shiftedit.net

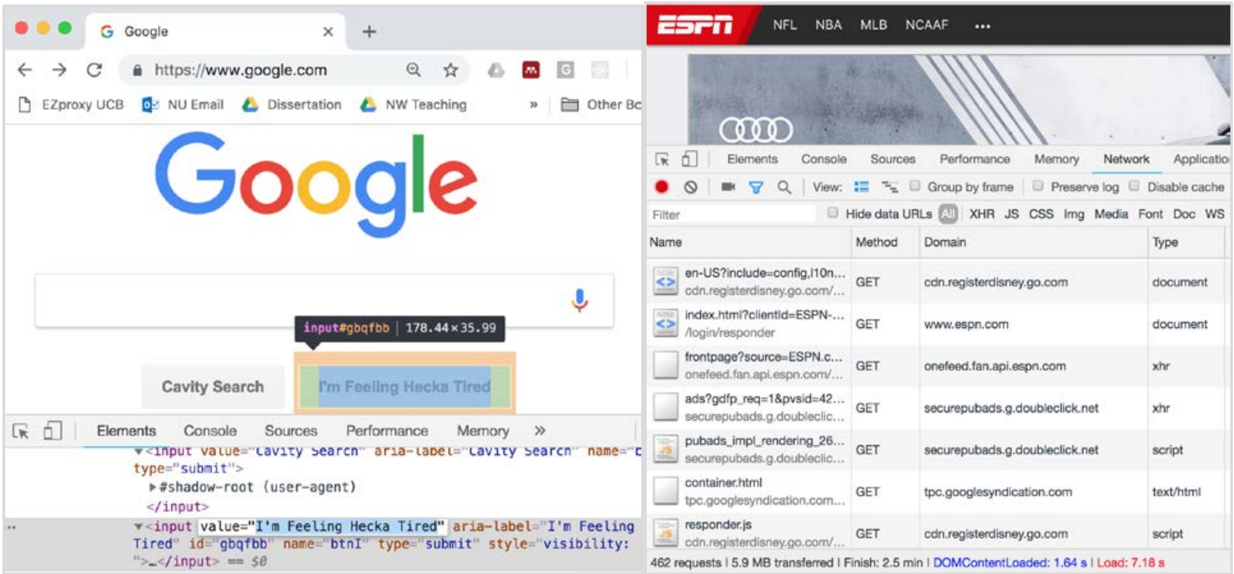


Figure 16: Chrome's "Developer Tools" being used to inspect and modify tags, and to inspect server requests

5.3.3 Starter code + tutorials

Starter code and tutorials were the primary means by which I tried to make particular aspects of web programming and architecture visible to students. Each tutorial started with underdeveloped but fully functional "starter code," and asked students to complete a series of modifications to the code to implement new features. Each task corresponded to a particular social or technical idea. Some tasks asked students to make use of an online code reference or practice using a debugging tool like the built-in developer tools in the Chrome browser. Other tasks required students to incorporate an external code snippet into their existing code – such as an audio player or a media feed. The goal of these tutorials was not only to help students develop content knowledge, but also to develop new practices with respect to reading and remixing code, debugging, leveraging built-in developer tools, and making use of online resources.

5.3.4 Code Snippet Catalog

My undergraduate research assistant and I also created a catalog of 75 "code snippets" that students could use in their projects (Figure 17). This repository had three purposes: first, to support students' understanding of how HTML, CSS, and JavaScript worked together to produce common features and functionality, including layouts, menus and slide shows. Second, to make available exciting (at least I hoped) interactive examples that students could incorporate into their computational artifacts. Some of the "flashier" snippets included animations, menus that zoomed in and out, image carousels, parallax effects, and other "modern" effects used by a number of popular websites. Third, we wanted this snippet repository to show the potential of working with other, third party internet resources like Bootstrap and JQuery. Each code snippet entry was linked to a pre-existing cloud-based teaching tool, CodePen,[31] that allowed users to view and edit the HTML, CSS, and JavaScript elements of any code snippet and see their results rendered on-the-fly. Whereas the code samples and tutorials were working programs in need of enhancement, code snippets were intended to provide the minimum amount of code they needed to support a specific feature or design.

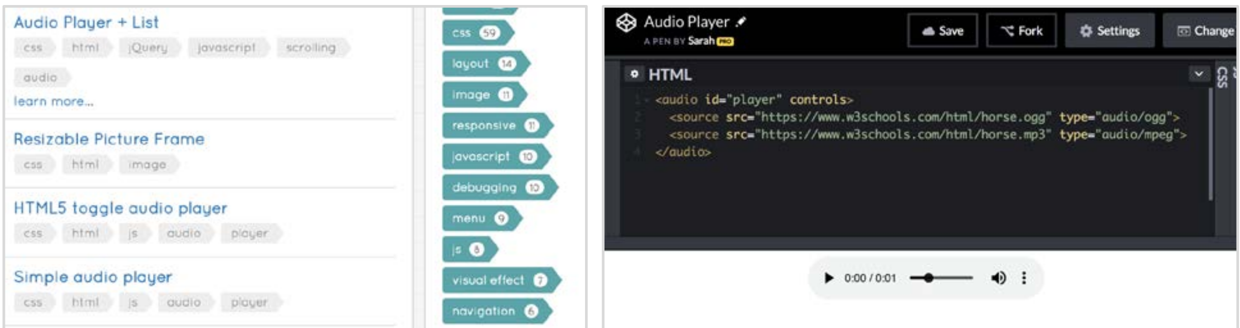


Figure 17: Browsable repository of HTML, CSS, and JavaScript code snippets. Each entry linked to a codepen.io sample that students could incorporate into their files.

5.3.5 API Tutor

I also designed and implemented a REST API querying tool, *API Tutor*, that allowed students to (a) browse the REST API endpoints made available by Instagram, Twitter, and SoundCloud, (b) view the JSON data feeds that they produced (c) query them; and (d) experiment with ways of visually displaying the JSON (using CSS, HTML, and a templating language to support binding data to the DOM (see Figure 1). Using *API Tutor* students would be able to modify the query parameters, endpoints, templates, and style sheets and receive immediate visual feedback (see Figure 18, Figure 19, and Figure 20).

5.3.6 Third-party Data Providers

Finally, I also asked students to register as authorized third party developers on Instagram, Twitter, and SoundCloud. This would enable them to generate API authentication tokens for their respective apps, and to read about the kinds of data that was available for them to create and to query.

---

[31] https://codepen.io

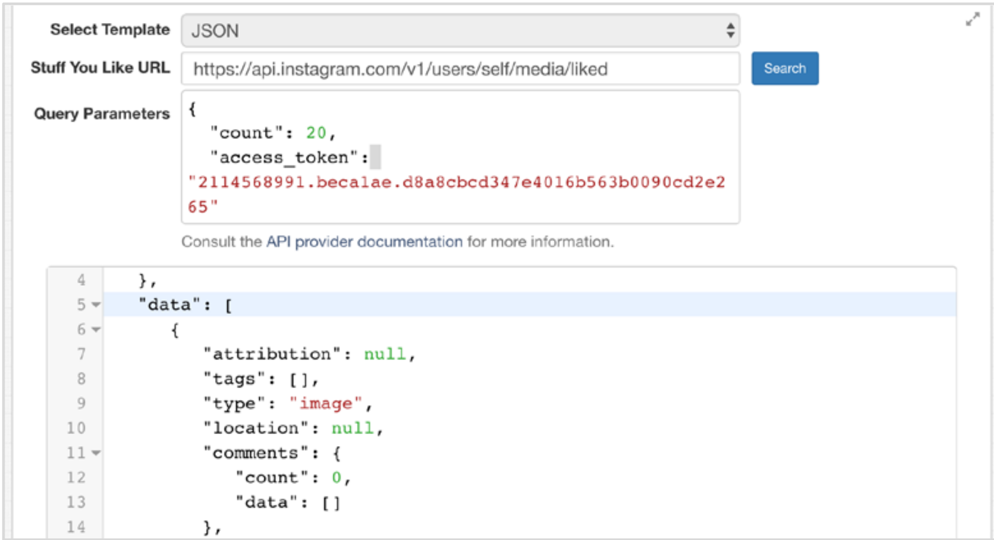


Figure 18: API Tutor data view, showing the JSON data returned from Instagram's "likes" endpoint

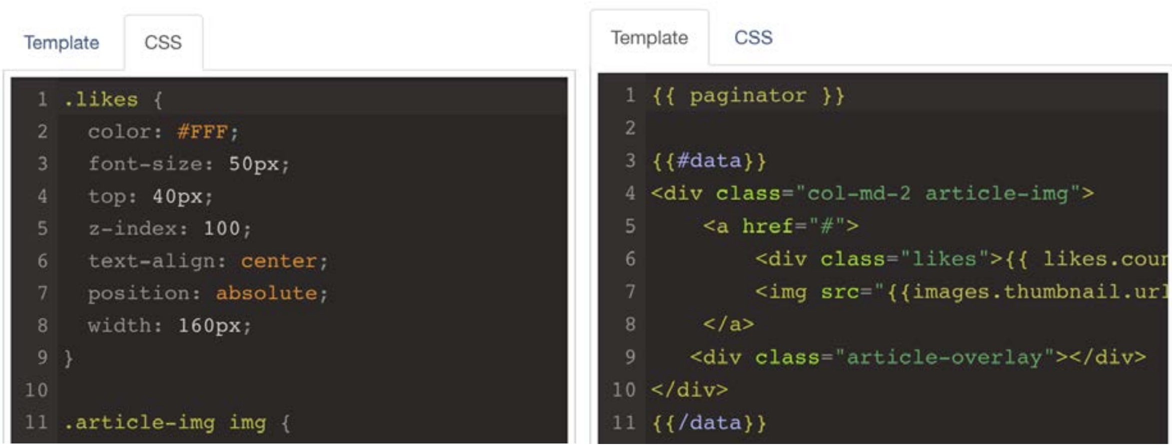


Figure 19: A template and CSS style sheet being applied to the "likes" data

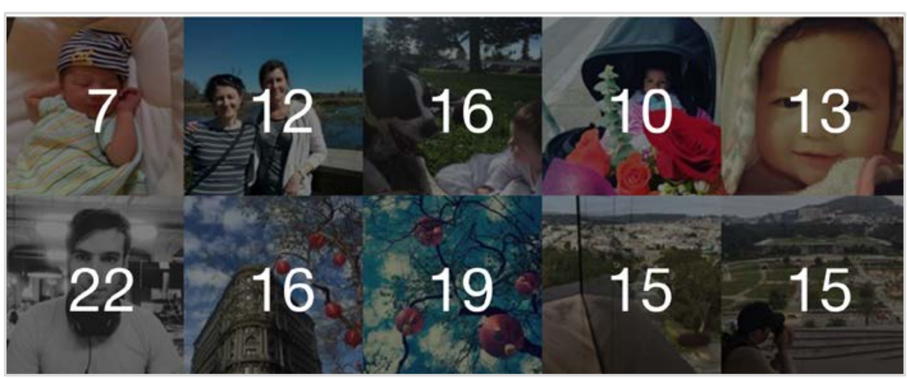


Figure 20: The resulting data transformation,
after the template and CSS stylesheet have been applied

5.4 Design Experiment Chronology

Over the course of five weeks (8 hours/week of in-class sessions and 2 hours/week of homework), the high school students participated in a series of hands-on activities, designed to help them consider some of the social and technical dimensions of networked computing.

5.4.1 Examining and building websites: Weeks 1-3

During the first half of the internship, the learning activities focused on examining and building websites. In the spirit of the use-modify-create pedagogy, students modified progressively more complicated sets of HTML, CSS, and JavaScript files – making a photo gallery, an animation, and eventually building a website from scratch. I also taught students how to use Chrome's built-in "developer tools" in order to examine the code of several popular websites (YouTube, ESPN, etc.). Students also explored the variety, and invisibility of various third-party scripts from advertisers and data brokers, using Chrome's "Network Panel."

Table 7: Computing practices and ideas explored during weeks 1-3

Activity	Computing-Related Practices and Ideas
Photo Gallery (Week 1)	- Exposure to HTML, CSS, JavaScript - Practice adding, removing, and modifying tags, attributes, and properties - Practice with related "vocational" skills (e.g., uploading, downloading, zipping, unzipping, saving, refreshing, naming conventions, etc.)
Hack a Website (Week 1)	- Exposure to Chrome Developer tools - Reading others' code - The idea that web pages link and embed disparate resources
Debugging Activity (Week 2)	- HTML: attention to syntax (e.g., opening and closing tags, nesting, using attributes); exposure to new tags and how to use them; debugging broken links and file paths; absolute and relative file paths) - CSS: attention to syntax and concept of selectors; exposure to new properties (e.g., color, layout, fonts) and how to use them - Practice reading others' code, tinkering, and considering common mistakes
Animated Cars (Week 2)	- Exposure to JavaScript syntax and some of its functionality (animation and scripts that continuously run) - Exposure to variables and loops
Website Redesign (Week 2-3)	- Design: analysis of good and bad examples of web design; exposure to basic design principles (colors, fonts, whitespace, contrast, hierarchy) - Coding: designing and implementing a web page from scratch; practice integrating code snippets into students' own code (via *Code Snippet Catalog*).

By the end of the first half of the internship, students had engaged in a number of technical practices, skills, and discussions; examined some of the underlying technologies of the web; and participated in several web design and web programming activities – including designing, modifying, and creating content, layout, styles, fonts, and colors. By having a baseline understanding of web languages, and experience with some key technical practices (modifying code, reading and incorporating code snippets, trial-and-error, downloading, uploading, saving, etc.), I hoped that students would gain technical proficiency and confidence, and start thinking about how they might apply their emerging competencies towards their own ideas and goals. These activities and the ideas and practices they motivated are summarized in Table 7.

Table 8: Computing practices and ideas explored during weeks 3-5

Activity	Computing-Related Practices and Ideas	Social Applications / Implications
Intro to API Tutor (Week 3)	- Exposure to queries (changing search parameters using URL queries) - Exposure to REST API endpoints; the JSON data format; the idea of entities, attributes, and relationships - Exposure to templates: how HTML and CSS work together to "transform" JSON into something viewable	- Exposure to the types of data that each service offered - Discussion around how the platforms may have obtained access to this data
Class Discussion: Platforms (Week 3)		- How do platforms make money? - Security + Auth Tokens - Logged into Instagram to see how many third-party apps had access to their Instagram. - Money, advertising, and privacy
Redesigning Instagram (Week 4)	- Reading sample code and examining some of the keywords and concepts in situ (variable, global v. local, assignment, etc.) - Exploring the template and modifying output (using loops) - Instantiating aesthetic effects from existing code - Connecting buttons to queries	- Registering as a developer - Concretely exploring *how* third-party data might be used.
Final Project (Week 5)	- Final project. Putting it all into action. Basic tenets: - Start from a template, understand how it works, and make it your own	- Exploring one's own ideas by making a custom third-party app - Performing / explaining how app connected to students' broader cultural communities.

5.4.2 Leveraging networked data: Weeks 3-5

During the second half of the internship, the learning activities focused on dynamically querying and manipulating networked data. Using API Tutor (5.3.5), the Code Snippet Catalog (5.3.4), and some starter code (5.3.3), students engaged in several activities that involved examining and leveraging data from Twitter, Instagram, and Soundcloud. I hoped that this approach would provide students with varied sources of data and inspiration for their final projects, and serve as a familiar context for thinking about how data are generated and used.

Specifically, students engaged in five activities: (a) working with the raw data feeds that each platform's REST API made available (using API Tutor); (b) deconstructing a simple web app; (c) registering as third-party app developers so that their web apps could leverage live data from SoundCloud, Instagram, and Twitter; (d) discussing how platforms operate and earn revenue; and (e) creating their own apps, their final project, on a topic that interested them – using HTML, CSS, and data from the various platform providers. For their project, I made some suggestions as to the types of projects that students might like to pursue, and provided some starter templates of working code, but students were free to pursue their own project ideas and discard the pre-made templates. At the end of the last day of the course, students presented live demos of their apps and PowerPoint presentations, explaining their design rationale, features of their app, planned functionality and next steps. . These activities and the ideas and practices they motivated are summarized in Table 8.

5.5 Methods

To explore the conjectures described in the opening of this chapter, I conducted a design experiment, which I carried out within a five-week summer internship program that I organized and hosted at the UC Berkeley I School. I recruited participants by contacting computer science teachers from two local secondary schools. I also recruited from *Code 510* (see 4.4). I received 22 applications from rising 10th, 11th, and 12th grade students, and admitted everyone into the program. 15 opted to participate, and 14 completed the program. Almost all of the participants had taken a computer science or game design course in high school. One third of the participants were female, and two thirds were male. Students' family educational histories were varied: 40% of students' parents had college degrees, and 40% did not (Table 9). Students' self-identified racial/ethnic affiliations were also diverse, and similar to the populations of the schools they attended (Table 10).

The internship required that students come to the university for two four-hour sessions per week and complete two hours of weekly homework on their own (a ten-hour weekly commitment, for five weeks). All participants were paid a $325 stipend and received a certificate of achievement at the end of the course. I was involved in all aspects of the program – including curriculum design, developing supporting tools, delivering instruction, designing materials and activities, leading discussions, and helping students to implement and debug their code.

Table 9: Highest level of education of parent(s)

Parents' level of education	
No high school	20.0%
Some high school	6.7%
High school graduate	13.3%
Some college / vocational	20.0%
4-year college graduate	13.3%
Advanced degree	26.7%

Table 10: Students' self-identified race/ethnicity

Students' race/ethnicity	
Latino/a	33.3%
2 or more races	26.7%
Asian	26.7%
African American	6.7%
White	6.7%

5.6 Data Collection

I video recorded all 10 sessions (class meetings), and wrote brief field reflections after each session, summarizing the notable events of the day. I later made video logs of each video, at 10-minute intervals, to create a coarse-grained transcript of each class session. For some of the programming activities, I asked students to record their screens using video capture software. There were some technical glitches, so not all of the videos were preserved, but each student managed to create a few. I collected all of students' written work, and had access to all of the artifacts they made through the course server. I also administered a pre- and post-course questionnaire, asking students about their knowledge of basic web ideas and their computing background. Finally, at the end of the course, my research assistant and I interviewed and audio recorded 13 of the 14 students (one intern did not want to be interviewed) to understand each student's perspective and ideas with respect to the course content.

To characterize student thinking about the implications and potential applications of platforms, as well as the growth in their technical knowledge, I chose to analyze five primary sources of data:

1. Video logs and field notes – to describe the course as a whole and students' engagement in the different activities.

2. Pre / post questionnaire data – to characterize what students knew coming into the course, and what they reported to be most interesting at the end.

3. Data corresponding to a class session on "platforms" during the fourth week of the course – to provide some context for what students were surprised by, and how both technical activities and whole-class discussions helped to support student thinking about these topics. These data sources include (a) screencasts of students registering as third party app developers and incorporating live data into their code; and (b) a transcript of a whole-class discussion about platforms and publicly accessible data.

4. Transcripts of 13 post-internship interviews, which capture some of the students' reflections on the course and their experiences. These interviews were conducted and by me and by two undergraduate interns.

5. A transcript of students interviewing each other at the beginning of the internship. This interview activity, which was intended to be an ice breaker, provided a way to share their interests, goals, ideas, and experiences with one another; and a way to help me design activities that could potentially resonate with students.

6. A transcript of 13 students presenting their final project. In these presentations, students described what they built, how they built it, why they built it, what they learned, and what they would do if they had more time. The transcript also captures the discussions and questions that followed each presentation.

7. Students' final web artifacts (i.e. the HTML, CSS, and JavaScript code they created).

The results of this analysis are presented in the following two chapters.

6 SKILLS, PRACTICES, AND PERSPECTIVES

In this chapter, I focus on the various competencies that students' developed within the internship, and how the learning environment (internship) mediated this process. To explore this question, I analyze two broad categories of competence: (a) students' engagement with computing- and data-related "concepts" and practices, and (b) their perspectives on the broader implications of web-mediated activity.

6.1 Technical Practices

As noted in 5.1, many introductory computing education efforts with middle and high school students involve making CS1 concepts (Tew & Guzdial, 2010) more intuitive by grounding them in contexts and activities that young people find engaging. This is achieved through student-centered languages and learning platforms, which are typically self-contained, use a more limited set of vocabulary, are often "block-based", and are application and platform-specific.[32]

While these efforts have had great impact,[33] I argue that these limited programming environments also run the risk of "abstracting away" aspects of the practice that are important for meaning and belonging. For instance, they may impact a students' ability to form an identity within the practice (Wenger, 1998) by limiting access to existing computing cultures and communities (e.g. online tutorials, videos, Q/A forums, venues for sharing and publishing one's work, code samples from which to learn, etc.). Moreover, it may not be clear to a student whether and how what they're doing with Scratch relates to what Google or Apple do if these connections are not made explicit.

These environments may also preclude students from developing a sense of competence as they engage in more advanced practice. By the time students get into high school, most are aware that becoming skillful in Scratch, Alice, and/or Lego Mindstorms does not have the same currency or authenticity as knowing C++, Java, or Python. Even if a teacher assures her students that they're learning "the same thing" as what they'd be doing in a real world language, there will always be an Nick in the class, who will remind his peers that C++ / Ruby / Objective C is what real programmers use (Weintrop & Wilensky, 2015a).

Given the potential limits of this sandboxed, "conceptual" approach to computing education, it seems important to ask when and where students are expected to learn the aspects that have

[32] That is, some learning environments focus on animation/storytelling or robotics or games or apps.

[33] For instance, studies have shown that student languages and learning platforms, such as Scratch, can indeed improve students' facility with programming concepts (e.g., loops or conditionals) as compared to a text-based language (Weintrop & Wilensky, 2015b).

been abstracted away, including how to express simple instructions using a formal computing language (Dorn, 2010; Nardi, 1993); organizing one's files and knowing where they are; using a text-editor; learning how to perform a query; installing and configuring software; leveraging online tutorials, videos, Q/A forums, and code samples; and participating in existing computing-related cultures and communities.

While professional text-based tools and languages can be complicated and tedious to learn, having the requisite support needed to use them can also foster important skills and dispositions that are essential to doing computing-related work (Guzdial, 2011, makes a similar argument). In the case of this internship, the practice of producing data-driven web pages required a set of skills that are not typically emphasized in introductory computing courses: understanding a directory structure (by understanding hyperlinking via relative and absolute paths), attending to text-based syntax rules, understanding servers, clients, and "the cloud," querying and sorting, filtering, and so forth. What learners got in return was access to networks, content – produced by friends, family, or strangers – anywhere in the world, audiences, and an endless library of coding blogs, samples, tutorials, fora from which to learn.

6.2 Social Implications

Engaging novice learners in web development also creates opportunities to examine some of the social implications of computing. Networked platforms and REST APIs enable valuable information and communication services (recommendation systems, localized search, connection to friends and family) precisely because they have access to massive amounts of personal data – collected, sorted, grouped, analyzed, and repurposed at an unprecedented speed, scale, and frequency. However, making personal data available to these systems[34] also makes individuals vulnerable to a variety of risks (e.g. surveillance, exclusion from critical material opportunities). Moreover, these risks are opaque, difficult to reason about, and disproportionately impact social groups who are already vulnerable (Eubanks, 2018).

Given this, fostering robust computing literacy requires helping young people to not only understand how these systems work, but also their implications. This includes supporting students to (a) think through some of the ways in which data and information are collected, repurposed, and commoditized; (b) consider how these systems support/erode important social values; and (c) ideally reflect on how these systems can be made better. As Gillespie (2010, p. 347) reminds us:

> Online content providers such as YouTube are carefully positioning themselves to users, clients, advertisers and policymakers, making strategic claims for what they do and do not do, and how their place in the information landscape should be understood. One term in particular, "platform', reveals the contours of this discursive work. The term has been deployed in both

[34] Many of which are owned by private, for-profit corporations and subject to relatively few governmental regulations.

> their populist appeals and their marketing pitches, sometimes as technical "platforms," sometimes as "platforms" from which to speak, sometimes as "platforms" of opportunity. Whatever tensions exist in serving all of these constituencies are carefully elided....as these providers become the curators of public discourse, we must examine the roles they aim to play, and the terms by which they hope to be judged.

Working with and leveraging networked content helps novices to concretely explore some of the possibilities and risks of networked computing, and could help them consider how different social values might be in tension with one another. For instance, how might the imperative to generate profit be in tension with combating fake political ads (Isaac, 2019)? Or, how does the value produced by gathering massive amounts of personal data compete with the privacy and security risks that are introduced? By allowing students to examine some of these tensions in light of their own data and systems, some of these considerations will emerge naturally on their own.

6.3 Methods

In order to examine some of the computing- and data-related knowledge that the learning environment did or did not support, I examine the following research questions:

1. *What* computing- and data-related competencies (concepts, practices, and dispositions) did students demonstrate over the course of the internship?

2. *How* did the learning environment support some of these emerging competencies?

To analyze these questions, I draw on student questionnaires, post-internship interviews and think alouds; students' final project presentation transcripts and artifacts, and some screencasts taken of students engaging in programming activities (data sources described in more detail in section 5.6). All of the interviews and presentations were transcribed by me and several research assistants. We also watched many of the screencasts together, and discussed interesting themes, particularly in regards to students' self-learning *practices*.

6.3.1 Data Analysis

To characterize the range of competencies that students demonstrated during the internship, I followed an approach similar to Brennan and Resnick's analysis of computational literacy within Scratch (2012). I began by looking at three categories of competencies as they related to the learning goals: (1) *concepts* – particular ideas common within computer science (e.g. conditionals, events, abstraction), (2) *practices* – processes of learning and doing (e.g. reading, debugging, integrating), and (3) *perspectives* – how learners "describe evolving understandings of themselves, their relationships to others, and the technological world around them" (Brennan & Resnick, 2012, p. 10). Next, the research assistants and I pulled out some finer-grained knowledge categories and examples within this high-level framework (Table 11). We also noted the activities and tools (i.e. context) that corresponded with these competencies to understand the role of various learning environment supports.

Table 11: Focal computing concepts, practices, and perspectives, with examples

Category	Sub-category	Example(s)
Concepts	Declarative programming	HTML: Adding, removing, or modifying tags, attributes, or values CSS: customizing properties and values
	The DOM	Nesting tags, using selectors to isolate elements
	Linking	Hyperlinking to other pages Embedding images and iframes Including external stylesheets and scripts
	Abstraction	Making a CSS class to control fonts and colors and reusing it Creating functions to issue different kinds of queries Recognizing entities, attributes, and values in JSON
	REST APIs	Modifying query parameters and values API keys and authentication
	Interpreting data representations	Examining data feeds and recognizing entities, attributes, and values Isolating particular entities and values within a tree
	Logical operations	Including "and" or "or" in Twitter queries
	Events	Linking buttons to functions
Practices	Debugging	Attention to syntax, error messages, guess and check
	Logistics	Downloading, uploading, unzipping
	Iterative programming	Guess and check
	Code integration	Finding examples on Reddit or from the *Code Snippet Catalog*
	Interpreting technical documentation	Learning about new endpoints and operators
	Using the developer tools	Inspecting HTML or CSS properties in Chrome
Perspectives	Privacy	Noting the kinds of metadata Noting the ease with which data can be searched
	Security	Revoking access to third-party apps Stealing someone's access token
	Money	Articulating the relationship between data and advertising
	Platforms	Interacting with APIs Linking to static images and audio files

6.4 Findings

During data analysis, the distinction between some concepts and practices was difficult to tease apart, particularly since each "concept" manifested in a particular practice that students were doing, and it wasn't always clear what constituted a stand-alone concept. As such, in the findings below, I decided to group concepts and practices together. Within the findings, I also note the particular contexts where a form of competence became salient, in order to make visible what may have been motivating and/or supporting the competency.

6.4.1 Skills and Practices

In this section, I describe some of the techniques and principles that students used to create their final project artifacts, how and why these techniques were used, and the kinds of emergent practices that enabled the production of these artifacts.

Web Programming

At the beginning the internship, most students had a notion of what HTML was (e.g. "I think it is based off the Internet and web pages"), two were able to articulate what CSS was (e.g. "a way to change fonts in coding"), and three could describe what JavaScript did (e.g. "a way to make pop ups") (see Appendix A, p. 142). By the end, every student was not only able to explain the purpose of HTML, CSS, and JavaScript (and how they worked together), but also marshal aspects of these languages in service of their final projects. While the projects varied in complexity, the extent to which students customized (or completely rebuilt) the final project starter templates (as noted in Table 12) revealed a basic understanding of declarative programming, as well as a number of other computational ideas.

Students utilized many different HTML tags, attributes, and values (e.g. iframes, images, links, lists, containers); and CSS properties and values (custom fonts, background colors, and layout techniques) in their final artifacts. These were largely driven by the desire to achieve specific aesthetic functionality, which pushed students to expand their language vocabulary, and to marshal tags, properties, attributes, and values in increasingly complex ways.

During the final presentations, when asked about the most challenging part of their project during the final presentation, four students responded with an anecdote about implementing a design feature. For example, Erick explained that he "wanted to have an immersive interface, so it's like a Parallax effect, and due to that, [he] had a lot of complications and tried to work around that." Sasha, who built his template from scratch, shared that "the hardest part was, first off coming up with the layout, because [he] didn't use a template, so [he] just placed all the text [seen on the screen]…all in CSS." Both Madeline and Jocelyn shared that getting their animated image galleries was the accomplishment for which they were most proud.

In 5 of the 13 projects, students completely abandoned the starter templates and introduced significant complexity into their code in pursuit of a particular design or feature. I observed students going to great lengths to figure out how to implement particular design ideas, sometimes spending hours positioning an image a particular way or getting an animated image carousel to work. Similarly, many of the technical discussions that took place during the final presentation also involved design features. Student questions included "How did you get them…them to be in boxes?" and "I was just going to ask if you bordered those two…everybody's really obsessed with your graphics. Did you blur the background of it too?"

Table 12: Summary of web and data features in final artifacts by student

	Declarative Programming			***Data Features***		
Student	Level of Customization	Custom Template	Custom Widgets?	# of Providers	Form of Query	Custom Queries?
Ada	Medium	N	Static Menu (HTML+CSS)	3	Search	
Michael	Medium	N		3	Search	
Nick	High	Y	Dynamic Menu	1	Search	Use of and/or conjunction
Anna	Low	N		2	Search	
Erick	High	Y	Parallax Effects	2	Browse	
Gabriel	Medium	N		3	Search	
Amar	Low	N		2	Search	Query by user
Jocelyn	Medium	N	Carousel	3	Search	
Angelo	High	Y	Dynamic Menu	3	Search	
Sasha	High	Y		1	Browse	
Teddy	High	N	Video Player	3	Search	YouTube integration
Derrick	High	Y		0	Browse	
Ling	High	Y	Dynamic Menu	3	Browse	

Web programming was also a concrete entrypoint into the idea of parametrization. In HTML, tags could be instantiated multiple times, where each tag would render its own unique element with distinct attributes, properties, and inner HTML. Parametrization also showed up as students worked with the data structures (entities, attributes, and relationships) that were generated and returned by the REST APIs that students leveraged. For example, SoundCloud's representation of the entity "song" always had the same attributes, but the values were different. Towards the end of the internship, students also learned to instantiate their own abstractions through named CSS classes, which could be applied to any tag. As students' CSS files became more unwieldy, most of them eventually bundled at least some of their repetitive style rules into blocks so that they could style multiple tags with the same ruleset.

Web Infrastructures and Data APIs

The learning environment also offered an opportunity to examine some ideas around web architecture, including an examination of how documents, media, and data are stored, accessed, linked, and searched across the world. To get at what students knew about the web coming into the internship, I asked them to define a few web-related terms in the pre-internship questionnaire – "the cloud," "web browser," and "data" (see Appendix A, p. 142).

From their answers I learned that while most students were aware of these terms/concepts, their ideas about them were a bit "hazy". For instance, most students indicated that that the term "the cloud" involved information/data storage. However, there seemed to be some confusion regarding *who* had access to data stored on the cloud (e.g. "where all data is stored for any and all to access"), whether or not the cloud was a *physical* or *intangible* place, and what kind of data and applications used the cloud.

Was it "a place to store pictures from your phone", or was it for "all data"? Regarding the term "web browser", most students confused it with either a search engine or a top-level domain. In regards to "data": students mostly described different properties of data – that it takes up space, that it is gathered, collected, and stored "in binary codes" and/or "in a base," and that "data is simply any information." Thus, while students already knew quite a bit, there were still some clarifications to be made – and particularly in regards to how data was stored and accessed in "the cloud."

At the end of the internship, five weeks later, I again asked students about their understanding of the web, and whether it had changed. Students had many answers to this question, relating to infrastructure, economics, privacy, security, the potential uses of networked data (addressed in subsequent sections of this chapter). Ada reflected, in her post-internship interview, that:

> [The internet is] not just a place. Like, it's not just Google and Bing where you just type stuff up. And there's, like, so much more to it. Like there's the server and then there's the browser and then there's, like, the cloud and everything because I know, like, back then people used to have, like, physical hard drives where they would upload to the computer. Now nobody has that.
>
> It was like a big bulky box. My dad has one, and his girlfriend's like *why don't you use the Cloud?* And he's like *No!* Like that movie with Cameron Diaz and, like, their video gets put into the cloud and they have to go break into Apple to try and get their video. It's called Sex Tape and they...I never saw it but I know the premise and they were like "The Cloud" and you can't take it (inaudible) when it goes up the Cloud. It's better to have your own hard drive I think.

She describes an entire infrastructure underneath the surface of the web, composed of servers, browsers, and the cloud, that had implications for privacy, security, and personal risk. Instead of storing things on a "big bulky box" that you could control, putting your files on the cloud was akin to moving your private data to someone else's computer. She connected these ideas to the film Sex Tape, a 2014 comedy that chronicles a married couple's efforts to destroy an embarrassing sex tape after Jay, the husband, inadvertently replicates it to various devices, including two iPads he had given away. Once it's in the cloud, Ada explained, you can lose control over it which could be really embarrassing.

Students also learned how to use cloud databases that were made available by various commercial platforms through REST APIs. Teddy explained the concept to audience members during his final presentation:

> Basically [an API is] the developers' tool that allows you to pull data from whatever, I guess, website that you want. So like Instagram, you pull data from there, and you can get your own account like this one…you can get data from your own account, and the same with like SoundCloud you can pull, like, your likes, your account information, just all of that.

By registering with each platform as a third party developer, students were able to integrate third-party data into their apps. Of the 13 final projects,[35] 12 made use of at least one API, and drew on a variety of different endpoints to instantiate their ideas (Table 13).

Table 13: Count of final projects using particular API queries

API Query	Count
User's own feed	10
Two or more endpoints used	10
Filtered by tag	9
Specific user's feed (hardcoded username)	1
Filtered by a series of tags joined by "OR" condition	1

Students also expressed an interest in connecting to other APIs that were not available through the API Tutor – including Spotify, Deviant Art's API, the New York Times API, and YouTube's API. A few students also asked if they could build their own API, so that they could host their own data, versus being tethered to a commercial provider.

Practices

The use-modify-create pedagogy fostered some important practices that were integral to students' learning process. These practices included things like engaging in trial-and-error strategies; reading online documentation; examining, integrating, and modifying code samples; and employing various debugging practices.

---

[35] One of the students was absent during the last week, and another exited the program in the second week.

Using Online Resources and Code Samples

One of the most generative practices was leveraging online resources and code samples. I tried to specifically design for this practice by encouraging students to inspect and modify examples as they were first learning the basic syntax and vocabulary of HTML and CSS.[36] Then, roughly two weeks into the internship, I introduced students to the *Code Snippet Catalog* – a repository of curated HTML/CSS/JavaScript code samples (see p. 68) – and asked them to modify and incorporate these samples into their own projects.

Thus, if the student opted to examine, say the "audio tag" example (which featured an HTML5 tag for rendering a simple audio player), she could inspect the structure of the tag to get a sense of the important attributes and the values that could be changed. She could then copy the tag into her own HTML file and change "src" attribute to point to an audio file of her choosing. In this way, each student was able to progressively expand their web language vocabulary while simultaneously enhancing their site.

Data from the screencasts also revealed that students' final project development process was largely a matter (a) looking for samples that were similar to the kinds of functionality they wanted in their app; (b) tinkering with the sample, and finally, if they understood it, (c) integrating it into their project. As Madeline explained:

> I found myself looking up a lot on the websites you gave us in the beginning, like how to do line breaks...that's really simple. But, over time, the simplest things, like the classes, the line breaks, the tags, all of that; putting images on: now I know how to do that. There are easily 30 things I can do, now that I remember it. So I feel like if I just do more, and make it even more complicated, which is probably how I'm going to learn more on the smaller little tags or whatever, is how I'm going to learn it I guess.

This process, however, was not always smooth. For instance, in one screencast, Jocelyn's search led her to a code sample for an image carousel that was implemented in PHP (a server-side web language that was inappropriate for the kind of programming that students were doing). In the screencast, she pastes the PHP code into her HTML file, saves it, and then refreshes the web browser to view the changes. Subsequently, her entire page disappears. Jocelyn responds by backing out her changes and repeating the same process in her CSS and JavaScript files. Eventually, she gives up on the sample and asks for help. Lucio (undergraduate mentor) is able to point her to a similar working carousel in the *Code Snippet Catalog* – which she eventually does get to work.

---

[36] To learn about HTML and CSS, students watched videos that reviewed syntax rules, modified and enhanced existing HTML/CSS files, and debugged common errors (mis-applied CSS selectors and broken file paths). See Chapter 5 for a more detailed account.

While integrating code from samples was not always successful, if a student didn't understand the sample relatively quickly they would either look for a simpler sample or else ask for help. Every final project artifact incorporated at least one code sample or template – either from the *Code Snippet Catalog* or from an Internet source like Reddit (Table 12). Students typically then made further modifications to these samples, including aesthetic and/or layout changes such as adding icons, fonts, colors, and images.

Students also consulted online documentation and manuals to gain a more in-depth understanding of the kinds of data that could be queried through the social media providers. When Ada asked Amar how he was able to display Lana del Rey's own feed, rather than searching for the tag "lana del rey," Amar explained:

> Oh, I went to the developers' [the API documentation], and I looked up the APIs, and there's an API where you can search by tag, and so I just put it into the HTML…no it was in the JavaScript…the Instagram JavaScript, and I just wrote the whole URL plus, you know, instead of yourself, you put, what it is. Hashtag whatever you want."

Another student, Nick, used the Twitter documentation to figure out how to search across multiple tags, using the "OR" keyword. In sum, as students competencies grew with respect to two core practices – reading, modifying and incorporating code samples; and interpreting online technical documentation – they were able to gradually expand their language vocabulary, and their repertoire of programming techniques, and their ability to interact with the online REST APIs in increasingly sophisticated ways.

Student Reflections

HTML & CSS

The use-modify-create pedagogy had both advantages and disadvantages. On the one hand, by the end of the internship, students were able to explain the purpose of HTML, CSS, and JavaScript and how the languages worked together. As Madeline reflected:

> I think I like what we did in this class because it allowed me to see how code can actually be in a final product. So even though I copy-pasted from lots of templates, you know, having a finished product and saying, this is what I can do in a couple of years when I figure out everything else, was, you know, very revealing. I was like, "Oh, this is what, how we can go further," or, I mean like that's what I can accomplish or even more, you know what I mean? So it was nice to have something to look at and be like, ok, like, in a couple of years, probably, I'll understand how to like, write it all myself.
>
> But, um. I dunno. I kind of just liked discovering it as I go along. I feel like ... if we did...I know it would help, probably, purely HTML. I don't know I think my brain would be fried every day. And I liked having the mix between the design and the HTML because, even just that little spark of color in the

> background makes me...more happier. I was like, oooohh! Yeah! Maybe that's just my personality, because I'm very artistic, but I don't know...maybe once I've had my little revelation about like, oh, yeah, I can actually use this code that we're learning, then I would probably want to go into detail.

As Madeline expressed, going into applications right away was motivating, and helped her to understand the purpose of learning the various attributes, properties, and tags. While she acknowledging that she copy-pasted from templates, seeing the big picture was important for her in terms of actually *wanting* to explore the details.

Memorizing language and syntax rules from reference materials was boring, even "brain frying." In fact, during the post-internship interviews, many students indicated that the instructional videos that went over specific HTML and CSS language features were their least favorite part of the internship. For example, when I asked Jocelyn what she wished we had skipped or spent less time on, Jocelyn said:

> Probably the HTML. It was informational, but there were just too many videos. Because I thought it was fairly easy […] and so after a while, I thought, "I know how to do this."

That said, all of them struggled to use the language features to produce the visions that they had for their pages. Jocelyn explains her experience with the assignment:

> **Jocelyn**: [Working on a web page from scratch] was really hard...I was really confused.
> **Sarah**: What was confusing about it? What was hard about it to you?
> **Jocelyn**: Um, like the layout. How to get the layout.
> **Sarah**: How to make it look good? Or how to position things?
> **Jocelyn**: How to position them and make boxes around the text. It was pretty challenging, but it was fun too.

As Jocelyn's reflection suggests, not only were the videos not particularly useful to the students, but they were also not sufficient in helping students apply the language features (tags, attributes, and declarations) to create something meaningful. Perhaps the videos (or some form of direct instruction) were a necessary part of developing facility with the language, but learning *to use* the language in practice was another skill entirely.

JavaScript

One of the biggest limitations of the internship from the students' perspective was that there was not enough time devoted to learning JavaScript. Everyone described the JavaScript activities as confusing or rushed. Despite this fact, one of the JavaScript activities involving animation was frequently cited by students as interesting and enjoyable. As Nick describes his thoughts on JavaScript:

> Yeah, um I kinda wished that we worked a little more on the JavaScript part of it, because I felt like that was probably the most confusing part. Like, I kinda understood the CSS and the HTML pretty quickly, but the JavaScript was like, "oh, that's pretty different. How do I do this again?...I really liked the JavaScript part of it, just because it was kind of new, and it was a little more complex than the other ones, and also you could do a lot of things with JavaScript, like you could make things hide and you can make things disappear out of nowhere, and I don't know. It was kind of a new twist to the entire HTML thing.

In their exit interviews, two-thirds of the students reported wanting to learn more about JavaScript when they were asked which topics they wished we had covered in greater detail.

Web Infrastructures and Data APIs

During the post-internship interviews, almost all of the students explicitly named learning about the third-party data APIs using API Tutor as the most interesting thing about the internship. Many expressed that they either hadn't really thought about it or else had "no idea" that platforms made their data available. As Derrick described, in his post-internship interview:

> Being able to share information the way you want to, is, like, information that you found in the way that you wanna use it. It's awesome! I definitely liked that the most out of everything [...] I really liked the data APIs, [...] and being able to, like, use like the entire Internet. Like, the entire Internet is at my disposal to like reuse and for other people to be able to view it conveniently.

From Derrick's perspective, APIs allowed him to access and curate the "entire Internet," exactly as he wanted. What was exciting to Derrick was that he could leverage APIs in creative ways, for other people. In addition to the creative possibilities that APIs enabled, students were also interested in the insight they provided regarding *how* social media platforms actually used data. As Michael expressed to Lucio (undergraduate research assistant) in his post-internship interview:

> **Lucio**: What were some of the things that you found interesting that you learned this summer?
> **Michael**: The data API thing because before I didn't know, like, how to do that at all. I had no idea how that worked, and then I came here and learned that, and I was like *wow* you can just pull stuff from social media and just put it on your page [...] I honestly don't think there's much difference with the apps that we make and the ones that we already have.
> **Lucio**: What do you mean the apps that we make?
> **Michael**: Like, like the feeds are still the same. I think the original websites have more functions. The ones that we made are just more simpler. Yeah.
> **Lucio**: And what were your favorite assignments in this class?

> **Michael**: I think it's the one that we're doing right now (the final project). Yeah because then like we're making what we actually want to make and what could be useful and what other people would actually use.

The APIs allowed Michael to make something that he "actually want[ed] to make" and "that other people would actually use." He was also able to equate the process with how other platforms actually created their user-facing apps.

6.4.2 Sociotechnical Perspectives

The internship also provided students with an opportunity to explore some of the social implications and potential risks introduced by social media platforms and the data they collected/disseminated, including privacy, security, and economic considerations.

Privacy

The issue of *privacy* came up repeatedly during the API-related portion of the internship. Specifically, as students began querying the REST API endpoints, they were able to see their data in each platform's repository and how easy it was to search. Moreover, there was an entire ecosystem of third-party applications – including the ones *they* were building – that leveraged this data. As Ada reflected in her post-internship interview:

> **Ada**: ...it's kind of scary what it can search up and what it can reveal, but it's also really cool what it can make.
> **Sarah**: Yeah it *is* kind of scary.
> **Ada**: Because I was looking at when we copied the two pictures from Twitter and it said locations of where this person was living and where they tweeted and I was like wow!
> **Sarah**: Really?
> **Ada**: Yeah it's // the girl was like from Brazil and the other girl was from Canada or something.
> **Sarah**: You saw that in, where did you see it?
> **Ada**: When we copied it to the document file it had like all of these different locations?
> **Sarah**: What, oh, just today? You saw it in the data?
> **Ada**: Yeah.

As Ada (and others) pointed out, data APIs were both "really cool," but also scary in terms of the kinds of personal information they could reveal. Examining the Twitter data feeds had enabled Ada to see two different instances where user's actual names and locations were readily available. Nick also noted the privacy implications of social media data as he was examining some of the customizable templates for displaying data within API Tutor (see p. 69):

> **Nick**: So [this template] (pointing to API Tutor) is only showing some of the information, it's not giving you all of the information. And um, it's

> actually…the other template wasn't giving you all of the information that you *could* see, but [...] I'm guessing...it's sending, like, the JSON, which I'm guessing is the same every time, but it's only displaying a certain portion of that information.
> **Lucio**: What do you mean only displaying?
> **Nick**: So this is basically only showing you the image and the number of likes, while the actual JSON has like, *oh, this is the location where it was posted, this is um, like the caption* that kind of stuff and all these other amazing things that no one needs to know because it's a crazy invasion of privacy. Laughs

By comparing the "raw" JSON data feed to the interface view that a user is typically presented with, Nick made an important distinction between "some of the information," as seen by the user, and "all of the information," as collected and made available through the API – characterizing the latter as a "crazy invasion of privacy" that "no one needs to know" about (said with irony).

Teddy raised another issue, pertaining to the *discoverability* of particular posts through the API's search interface. As he describes:

> **Sarah**: What were some of the things that were more interesting and that you remember learning [...] anything that kind of stands out to you that you remember?
> **Teddy**: I think honestly the API thing was the coolest thing. Just being able to see how things work behind the scenes of the bigger things that people have made it was really cool.
> **Sarah**: Say more about it. Like what? Just the business aspect or the industry aspect or the technical aspect?
> **Teddy**: Definitely the technical. Just seeing how everything is connected with the API and just how you can find something so, like, so specific. It's just kind of interesting but kind of scary at the same time.

Through the API, Teddy had seen, in a new way, how "everything was connected." Not only could you see your own data and that of your immediate networks, you could see anyone's data, if your privacy setting was marked as "public" (whether intentionally or unintentionally), by simply issuing a targeted search on a keyword or username.

Security

Concerns around "getting hacked" came up while students were registering as "third-party developers" on Instagram, Twitter, and SoundCloud. In order to access user data, most platforms require that the developer first register with the platform using the OAuth2 protocol. Each student had to log into all three platforms (Instagram, Twitter, SoundCloud) and fill out a form where they were asked to describe the purpose of the app they were building and the level of data access they wanted.[37] After registering, the platform issued each student a unique "client id," which they had to add to their app's code in order to be able to query the platform's data. This client id also allowed the platform to monitor how each third-party app was using its API, ideally to ensure user safety and that the third-party app was complying with the platform's terms of service.

The OAuth2 model also had a second, user-facing step, which many of the students had already encountered in their daily social media practices (often without ever remembering). This step required that the user give the third-party app permission to use her data, prompting her with a message like:

> Do you Authorize **{{ name of third party app }}** to access your Instagram data with the following permissions **{{ read, write }}**?

If the user allows access, then Instagram will give the app the user's "access token." At this point, the app could potentially save the user's access token indefinitely until the user manually revoked access or until the token expires. Research has shown that few users really have any understanding of the access they have granted to a third party agent (King, Lampinen, & Smolen, 2011; Rosenberg, Confessore, & Cadwalladr, 2018). Even developers also struggle to understand how to correctly handle security via OAuth (Chen, Chen, Tague, & Kotcher, 2014).

As I reviewed this workflow with students, several questions came up. For instance, Sasha asked: "Can they log into your Instagram anytime they want?" And Ling followed up by asking: "Do they know your password?" I explained to students that the power of OAuth2 was that you didn't have to give third party apps access to your password. I also (naïvely) assured students that third party apps had an incentive to act ethically lest they be banned from the platform, and that it was usually fine to allow these apps to access their data.

Immediately following this discussion, all of the students (unprompted by me) logged into their social media accounts in order to view all of the third party apps that *they* had at some point authorized to access *their* data, and at what "level" of permissions. Several of the students began going down this list and revoking access to all unfamiliar third-party apps (Figure 21) – many of which they had no memory of authorizing.

[37] We glossed over any discussion of the implications of these various choices – treating it more as a technical exercise to get through in order to get back to the coding. However this was a missed opportunity.

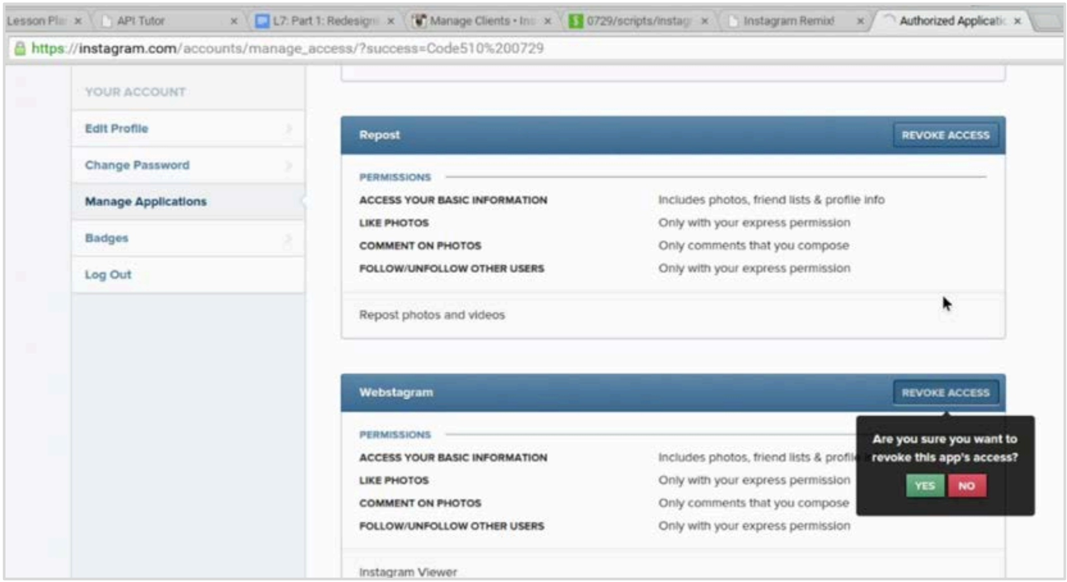


Figure 21: Ada, who is logged into her Instagram account, reviews the list of (four) third-party apps to which she has given access to her data, and revokes their access

Another security-related insight emerged from a "bug" in my code. I had accidentally included my *personal* Instagram access token within the "starter code" for the day's activity. As a result, all of the students' were able to access my private Instagram data. As Madeline recalled in her post-internship interview, it was in this moment that she realized that she could be a hacker too:

> "[In school] they were explaining how much you have to be careful about what you use on the Internet, and they were just scaring us basically with all these statistics. And I was a little scared...but then afterwards, I was like, "Whatever!" But now, like, oh! I can actually do it. Like, right now probably! Just the knowledge now of actually being able to do it. I always just thought, like, these really, really smart, elite, hacker people could do it. And I was like, "Nope." Like if you just know it. Like if you're taught, that...it's...probably you could do it too.

Madeline was able to see for herself, through a bug in my code, how a bad actor might go about accessing / stealing data from another account, even one with the most restrictive privacy settings. "Hacking" wasn't about being a computer supergenius – it was about understanding how data access and authentication worked, and "if you're taught, you could probably do it too."

Economics

Another source of curiosity involved understanding how some of the web giants actually used data and information to generate revenue. At some point during the internship, 11 out of the

14 interns reflected that they didn't have a solid understanding of how these web giants made money. Consider the following three reflections:

> **Jocelyn**: Like...when we did the things with the companies and how they make money, that was really interesting because I didn't know that at all. Like the ads. Um, I never knew they charged you...well I've never clicked on an ad, but I didn't know they charged you or anything. And I didn't know how they made money. I was actually really confused about that. I didn't know how Google made anything!
>
> **Madeline**: I had no idea how Facebook and Twitter – I still don't know how Twitter gets money. How does Twitter get money? And YouTube too – how does YouTube get money? Ads too, but they have to pay people too, like those really good YouTubers...they get paid. It's like a job.
>
> **Derrick**: Like when we were talking about business and how people make money on the Internet, I kind of knew that but I didn't, like, exactly know how. Like all these people are making a lot of money each day, every hour actually and it's like very lucrative. And I don't know. I'm just fearful for the internet because it could become like, issue. Net Neutrality now is like really big so we're gonna have to, businesses are gonna start taking over the internet and its gonna be defeating the whole purpose of it actually. Its for anyone to access it.

By the end of the internship, many of the students shared that working with data helped them to better understand how companies made money off of code. For instance, in the exchange below, Erick shared not only how targeted advertising worked on Facebook, but also how ads could become even more targeted and specific:

> **Sarah**: Ok so umm do you think your understanding of the Internet has changed at all since the internship?
> **Erick**: Yes. Like the part where, like, money is involved. Like realizing that a lot of companies make money, like, off a lot of things like coding stuff and things.
> **Sarah**: So how do you think it's getting those particular ads onto your Facebook? Like how did that happen?
> **Erick**: I guess it's like matching up tags and stuff like that. That have, like. If I like kittens, then there's like an ad for a kitty, like, selling business. It, like, clicks onto me and then it's like: "Oh he likes cats so gotta get that cat ad" kind of thing.
> [...]
> **Sarah**: Do you think that Facebook has saved every single thing you've ever posted?
> **Erick**: {{nods}}
> **Sarah**: Yeah? umm and do you think they've done that with every single user they've ever….

> **Erick**: maybe yeah
> **Sarah**: mhmm so what you think it means to have every single person's...looking at maybe half of the world's information...in a single database owned by one corporation?
> **Erick**: Yeah it's pretty scary.
> **Sarah**: Yeah it's pretty weird huh? What's scary about it to you?
> **Erick**: it's weird and scary to me, like, cuz it knows pretty much everybody who's online and, like, all their stuff is in there sort of thing.
> **Sarah**: So how does Facebook make money?
> **Erick**: ads? Yeah.
> **Sarah**: So in the future, do you think there are other ways they could use all that data to make money?
> **Erick**: I guess more ads (laughs)? Like more really specific ads that, like...say you would post, like, a status that says "my car got like crashed" or something and then Facebook looks at those keywords and then shows you random ads for places where you can get your car repaired and then you buy a new car and stuff like that.

In another post-internship interview, Nick brought up how online popularity was also commodified:

> **Jessica**: And how do you pay for the internet?
> **Nick**: So YouTube, I think it gets most of its money from Ad revenue, so you don't really have to pay for it. Well technically you are paying for it because you're viewing the ads and the ads are paying YouTube and that kind of stuff, so it's free.
> **Lucio**: So how does, how does YouTube make money off of you? Like how does it use your views? Or does it not?
> **Nick**: So, very popular videos probably get a lot more ads around them because more people are seeing them. And users with a lot of subscribers will have a lot more ads because people like putting their ads on things that get a lot of views. And so if they're putting the ads there, they're giving money to YouTube to put their ad there. So they get people to view their ads. OK. A lot of ads.

6.5 Discussion

In the findings above, I describe how a particular kind of learning activity – which involved using real-world web languages and third-party data feeds to produce web apps – helped students to learn about computing, data, and web infrastructures. In this section, I consider what these findings might mean for the design of learning environments that aim to support computational literacy among high school students.

6.5.1 Coding with Real World, Declarative Languages

Learning by example

The logic of authoring web pages appeared to be accessible to the students (after getting over the initial learning curve), perhaps because HTML and CSS are high-level, declarative languages with fairly simple syntax and composition rules. However, learners also need to (a) *want* to understand how the tags, attributes, and style directives might be assembled, and (b) have the opportunity to study how this might be done through examples and libraries.

With respect to *wanting* to understand, having engaging examples – music players, carousels, animations, beautiful interfaces, clean designs – sparked students' curiosity. Through the process of examining and modifying the attributes, properties, and values from the examples, their programming knowledge grew. Much like other studies of programming practice – whether in the context of professional graphic designers (Dorn, 2010), accountants using spreadsheets (Nardi, 1993), or children using the Scratch programming environment (Dasgupta, Hale, Monroy-Hernández, & Hill, 2016) – remixing and learning by example appears to be a productive vehicle for expanding high school students' programming repertoires for the web.

Skills and Practices

Learning environments could also spend more time centering the development of *practices* that enabled students to learn by example. These include learning to search for and integrate code samples online by swapping out parameters and arguments and debugging the result. Learning by example still requires that the learner have a baseline understanding of the syntax rules and vocabulary of the language. I argue that the structured "skills" activities made the learn-by-example practices productive (particularly for those who were completely new to the medium). While some activities are boring and tedious, that doesn't mean that they aren't valuable and necessary, even in learning contexts that are focused on creative production and play.

Real-World, Text-Based Languages

These findings also suggest that students were willing to learn these tedious and boring syntax rules and work through bugs and confusion when they perceived their efforts to be worthwhile. I argue that their assessments of worth were related to the real-world distribution channel and aesthetic expressivity afforded by web tools like HTML, CSS, JavaScript and REST APIs. Using real-world languages also gave students access to a wider distribution channel, and a wider availability of tutorials and sample templates. This enabled students to create sites that looked modern, professional, and aligned with their aesthetic vision.

6.5.2 Web Infrastructures and Their Implications

While an examination of the social implications of platforms was not the primary purpose of the internship, questions and concerns around privacy, security, and money nonetheless came up over the course of students' app development practice. Working with the REST APIs and raw data feeds enabled students to see the personally identifiable and potentially "revealing" metadata associated with each and every post, tweet, or upload (e.g. time stamps, location information, key words, full names). They were surprised how easily these data could be searched by *anyone*, whether they knew them or not.

6.6 Conclusion

By the end of the internship many students had a better grasp of how the web worked, how some of the pieces fit together, and had developed a variety of generative, cross-cutting technical practices relevant to web development. These learning outcomes were enabled by designing tools and activities that tapped into students' existing, online cultural practices. I argue that this approach to introducing computing-related ideas and practices – where students have the opportunity to explore their everyday media ecologies in terms of their underlying computational principles and implications – can provide an intuitive and motivating context for learning computational skills.

The web *already* serves as an important medium of cultural production for youth. Because the internship prioritized learning by example, using the starter code and snippet library, students were able to explore how this cultural production happens, and how expressive it can be, for themselves. By helping students to interoperate with third-party data APIs, the internship helped them to examine how data, infrastructure and design connect to create user experiences on the web. Lastly, the internship helped them to put all of the pieces together to perform their own acts of cultural production. "Even though [they] copy-pasted from lots of templates," (Madeline) students valued understanding the process for making a finished product, and recognized that their final products were "not that different" (Michael) from commercial social media apps.

7 IDENTIFICATION AND BELONGING

As Wenger (1998) argues, learning takes place as people seek out new identities and modes of belonging while participating within a community of practice. Through this participation, a person can "explore new ways of being that lie beyond [their] current state" (Wenger, 1998, p. 263), and eventually come to understand the world – and their relation to it – in a new way. In this view of learning, the pursuit of meaningful participation is primary, while the development of new skills and practices (typically the purview of educational initiatives) follows. That said, acquiring new skills and practices also allows people to move into new situations, which can allow them to take on additional identities, which can motivate them to develop new skills, and so on. Thus, the pursuit of meaning and identity is expansive and transformative.

Given the primacy of meaning and belonging, this chapter examines some of the ways in which students identified (or did not identify) with the practice of computing. I begin by taking stock of what computing meant to students prior to the internship, drawing on students' reflections and descriptions of their everyday computing and media practices; their experiences learning about computing in computer science class; what they expressed wanting to make, do, and learn during the internship; and how they saw computing figuring into their longer-term trajectories and identities. I organize these reflections/descriptions according to the three modes of belonging proposed by Wenger (1998): *imagination*, *engagement*, and *alignment* (described below).

After establishing this baseline, I then go onto examine some of the ways in which students engaged in web development, as well as some of the meanings that students assigned to the practice. I do this by noting the different kinds of apps that students built, how students related their apps to their broader interests and values, and the different trajectories of engagement that students pursued within the practice.

Identity and Belonging

In order to examine what the practice of computing meant to students, I focus on the concept of **identity**. *Identity* links the individual and the social by considering communities and social categories in terms of *belonging*. As Wenger explains, "talking about identity in social terms is not denying individuality but viewing the very definition of individuality as something that is part of the practices of specific communities" (1998, p. 146). Moreover, it brings into focus the impacts that different kinds of social processes (e.g. economic structures, power, social norms, etc.) might have on an individual and vice versa (see section 2.1).

Wenger argues that education should help people explore new identities within a practice, rather than solely focusing on skill-building and information delivery activities (which tend to be highlighted in K-12 educational settings):

> Whereas training aims to create an inbound trajectory targeted at competence in a specific practice, education must strive to open new dimensions for the negotiation of the self. It places students on an outbound trajectory toward a broad field of possible identities. Education is not merely formative – it is transformative (p. 263).

If helping learners to explore new identities is the goal of an educational initiative, then the question becomes: how this might be done? Wenger argues that it is important to attend to three distinct but interrelated modes of belonging in order to foster identity formation within a practice:

1. **Imagination**: One's ability to create images of what a particular practice might influence, affect, or make possible. This includes projecting new images of oneself and of the world. As Wenger puts it, imagination is "playing scales on a piano and envisioning a concert hall. It is reading a biography and recognizing yourself in the struggles of a character."

2. **Engagement**: One's immediate participation within the social practice. Examples of engagement include doing work that is valued in the practice (i.e. developing competence), forming and maintaining relationships, following rules, showing up on time, and so forth.

3. **Alignment**: One's willingness to do one's part to contribute to the broader enterprise around which the community of practice is organized. While imagination is what enables a person to project new images not directly experienced, alignment is *actively choosing* to dedicate personal energy and resources towards an image of what the practice makes possible.

Each of these modes of belonging works together to promote identification. As such, learning environments that privilege identification with a practice must help learners to do all three: understand what the practice makes possible (imagination), develop competencies which are valued within the practice (engagement), and see how their participation in the practice relates to the envisioned outcomes (alignment). These concepts can also be used to analyze past learning experiences in terms of the kinds of opportunities for belonging that were present / absent within a particular practice.

To examine the process of identification and belonging for the high school students who participated in this study, I draw on data related to students' direct participation in the internship activities, as well as students' retrospective and prospective accounts of their relationship to computing. In doing so, I ask three research questions:

RQ1: How did students identify with the practice of computing prior to the internship, given the three modes of belonging described above? This includes why computing knowledge mattered to them, what they hoped to gain from the internship, and the kinds of knowledge and goals they opted to pursue during the internship.

RQ2: How did students identify (or not) with the practice of computing during the internship?

RQ3: What were some of the resources and constraints (both within and outside of the internship) that encouraged / limited students from developing a *sense of belonging* within the practice of computing?

7.1 Data Analysis

To analyze some of the ways in which students identified with computing, I drew on three data sources: (a) transcript of students interviewing each other at the beginning of the internship, (b) a transcript of the final project presentations, and (c) transcripts of the post-internship interviews (data sources are described in more detail in section 5.6). Using these transcripts, I engaged in several rounds of coding, looking at two scales of activity – computing as a broader cultural phenomenon, and computing in relation to some of the specific internship activities.

Computing as a Cultural Phenomenon (RQ1). I began examining the transcripts by first attending to the meaning that students assigned the practice of computing broadly. Why had students come to the internship? What about computing seemed important to them? Were they planning to go into the tech industry or to pursue further studies? What were some of their prior experiences with computing? Many themes emerged from this initial pass: anxiety around accessing opportunity and falling behind; curiosity and confusion; experiences of feeling lost; an interest in making something that others would find and appreciate; a desire to figure out where students were headed, etc. To further deconstruct these themes, I grouped these codes according to Wenger's "modes of belonging" (engagement, imagination, and alignment). Findings from this analysis are described in section 7.2.

Computing within the Internship (RQ2). I also examined meaning and belonging as it directly related to the internship activities, using students' final project artifacts and presentations as one window into understanding these ideas. What did they build, and why did they build it? Here, I coded for the kinds of trajectories that students pursued, the interests and values they embedded in their apps, and how students talked about what their apps meant and how they related to them. Findings from this analysis are described in section 7.3.

Mediation (RQ3). In addition to naming how students identified with computing, I also wanted to understand some of the factors that were shaping the process of identification. Therefore, I also looked for various mediators of belonging (at each scale of activity). At a broader scale, some of the emergent themes included: narratives about the future; prior knowledge; parents; interests; values; etc. At the scale of the internship, additional themes included content, communities, and design. Findings from this analysis are described in section 7.4.

7.2 Retrospective and Prospective Accounts of Belonging in Computing

Students brought many interests, expectations, and beliefs into the internship, which shaped how they related to the internship and to the practice of computing in general. Some of these came from prior experience, both within and outside of school. Others came from parents, teachers, the media, and broader cultural narratives. Drawing on students' pre/post interviews, I group these accounts according to Wenger's three modes of belonging (experience, imagination, and alignment). While self-reporting and self-reflection are not without limitations (e.g. reliance on people's partial memories, the difficulties in making subconscious processes explicit, etc.), these accounts nonetheless provide an important (if incomplete) window into how young people come to know computing-related ideas, and how we might better support them.

7.2.1 Imagination

As discussed briefly in the beginning of this chapter, imagination is what enables someone to envision how a practice matters in the world, and what one's participation in the practice might look like. Or put another way, imagination is "a process of expanding ourselves by transcending time and space, and creating new images of our world and ourselves" (Wenger, 1998, p. 176). Imagination can motivate a person to enter into a new practice by helping them form an image of how it might become meaningful. However, it is important to note that these imaginaries do not have to be true in order to foster a sense of belonging- they only have to be plausible to the person projecting them.

For example, a person working at Facebook might imagine that the feature they've created will make the platform easier to use, thereby making the world a better place, irrespective of whether either of these propositions holds. Moreover, because stereotypes, misconceptions, aspirations, cultural scripts, and experience can each inform these images (Wenger, 1998, p. 177), imaginaries can reveal the kinds of influences to which people have been subject. Given this, as I describe some of students' imaginaries below, I also highlight some of the cultural narratives and kinds of prior experiences that these imaginaries potentially reveal.

Imaginaries for the Internship

At the beginning of the internship, students shared a variety of computing-related images, which related to the kinds of things they wanted to build, learn, and participate in during the internship (and presumably beyond it).[38] For instance, as Gabriel shared with Ada:

---

[38] They did this during an ice breaker activity, where students interviewed each other according to the following prompts: (1) what computing technologies do you regularly use? (2) What do you want to potentially make during the internship? and (3) What do you want to learn? Each student interviewed 5 other students, and each interview lasted for 5 minutes.

> I want to do so many things that I don't know what to do! I want to, like, create a game, like a digital game that like…you know like Crossy Road or something, I don't know something like easy but like, something that's addicting, but not as hard to make...And then I want to make a website…I don't know of what. I just like to design stuff, so I would like to design a webpage. And I would like to create an app that gets Instagram, Facebook, and oh, Snapchat all together into one...

Notably, each of Gabriel's ideas is rooted in a digital media platform / website with which he already had familiarity but believed he could improve upon (e.g. by creating an integrated social media platform). Similarly, Jocelyn had a game idea, which was based on a meme involving a rapper, Rick Ross, who gained national attention when he lost 100 pounds by (allegedly) only eating pears. As she explained to Madeline, she wanted to create a game that involved aiming and flinging pears into an animated Rick Ross's mouth, causing him to become skinnier as the game progressed. She believed that her idea had the potential to go viral, and perhaps even catch Rick Ross's attention. Like Gabriel, Jocelyn imagined producing her own take on a cultural phenomenon using skills that she hoped to learn during the internship.

Students also described the kinds of *qualities* that they hoped their project would have. One important quality was aesthetics. About half of the students explicitly mentioned that they wanted to build something that looked really "clean," "polished," "awesome," "professional looking," or "pleasing to the eye." As Nick shared with Ada, "I want to learn in this class is how to make a website that looks fabulous, you know, just like Apple."

Another quality mentioned by several students, was the idea that something they built could make a unique contribution "in real life." As Jose explained to Gabriel:

> Remember that game we keep playing at school? {{Inaudible}} That shit is hella fun! And then I want something that…I want a game, but I also want something you can use on a daily basis in real life, like an app…like the app I have to find cops….laughs…like a police scanner. I don't know. Anything. Something that's helpful and that people would use on a daily basis...and I want to make my own thing and say it was mine, basically. You know, something that I produced, that people use and find.

Whether a game, a police scanner, or some other pursuit, Jose wanted to create something that was uniquely his – something that not only reflected his own knowledge and ingenuity, but that was also appreciated by and useful to others. Moreover, by being able to "say it was mine," Jose gestures to the importance of recognition, and the potential credibility that might come from creating something genuinely useful or fun.

In sum, students had many images of what they might create and explore during the internship, influenced by their daily interactions with various technologies, platforms, and cultural forms. While some of these ideas were very specific (e.g. Gabriel and Jocelyn), most simply hoped that their projects would align with particular qualities that were important to them: being used and appreciated by others, "looking really awesome," and/or having the potential to make a broader impact.

Future Imaginaries

Students' also shared ideas they had about their future – both how they might navigate a future world mediated by technology, and how they might like to participate in that world. One future consideration was of course the college application, which was a significant (if not primary) motivator for many students to participate in the internship. As Michael expressed: "I think [the internship] would look good on my college app, and my computer academy teacher said it's a really good internship here." Madeline expressed a similar sentiment:

> UC Berkeley (the sponsor of the internship) just sounds so good on your resume. You know, that just looks really good!..and I know this is just my mom talking, but it would be nice if we could get a certificate at the end of this (laughs)...

Students (and their parents) were keenly aware that a STEM internship had other kinds of currency, including making them more competitive in the college application pool. That said, future imaginaries extended beyond college admissions. Another recurring theme expressed by students was an abstract notion of "being prepared" for a future increasingly mediated by technology. As Madeline put it:

> I just know *that* (emphasis mine) knowing computer science might help me in the future. Because I feel like, again, this is gonna...in years to come, technology is going to consume us, and knowing it definitely will help somehow..maybe in the simplest ways, I don't know.

However, when I asked students to name a potential situation where computing might be useful in the future (beyond procuring a tech job), most students couldn't. For instance, consider this exchange during my post-internship interview with Jocelyn:

> **Sarah**: Why do you think it's valuable to learn about computing? What kinds of things does it help you with?
> **Jocelyn**: Um....I guess.....(pause). I'm not sure.
> **Sarah**: So you said at the beginning [of the interview] that you know tech is everywhere and that it's the future. Like, what do you mean by that...when you're thinking of tech and the future?
> **Jocelyn**: Well I know there are a lot of advances coming out and I know that eventually we're going to be surrounded by technology and if you don't know how to use it, then you're just (pauses). It's going to be really hard.

Perhaps, then, students were hedging their bets. Maybe they weren't necessarily that interested in computing per se, but in the opportunities (albeit abstract) that computing skills might open up or preclude? This idea certainly aligned with the overarching narrative undergirding many

CSforAll initiatives: that there were tremendous material opportunities associated with having computing skills; not learning these skills would put someone in danger of being left behind.

I also asked each student during the post-internship interview if there were any specific careers they were considering that might involve computing. Collectively, students named a fairly narrow range of career options, which involved either working for Google, or making video games, animations, or robots. In Teddy's post-internship interview, he mentioned being curious about how Google worked from the inside:

> **Teddy**: I really think that technology is the future, so I want a career in that somehow just...I want to make something, maybe, that people will, like, just create something completely new but also I kinda wanna work for one of those big companies just so I can see how they work and see the inside of it.
> **Sarah**: Uh huh. So name a company that seems interesting to you.
> **Teddy**: So definitely Intel or Google – to see all these things that we don't see but are there.
> **Sarah**: Like what's something that you think seems interesting?
> **Teddy**: Just like all the data that goes through Google I would wanna take a look at that 'cuz it just seems so cool like just all the data from around the world that goes through Google just seems kind of like it's not you don't think about it until you see it and then you're just like wow that's a lot of data!

Jocelyn also named Google, explaining that "it just seems really fun. Like a really fun environment....I guess it's because of the movie, T*he Internship*.[39] It just made it look really fun."

About half of the students expressed that they did not want to pursue a tech job. Amar shared that he wanted to be a doctor, and suggested that perhaps computing knowledge would help him troubleshoot the medical devices he would no doubt rely on to treat patients. As he explained:

> So...if there's like a serious concern (with some medical machinery), I can fix it, like, instead of having someone who's like, kind of like a technician or something come and fix it. And that'll take, like, hours and I have, like, a serious patient!

Ada didn't know exactly how computing skills mapped onto her future aspirations, but liked the flexibility that computer science seemed to offer:

> I'm still figuring out what my interests are but for me a passion is that like, I can apply to anything I do. It like, if I wanted to pursue fashion, design I can still use computer science, like, you can always use computer science every single like almost every single job you do.

[39] A comedy about Google starring Vince Vaughn and Owen Wilson.

Madeline, on the other hand, was less satisfied with the assertion that computing knowledge was generally valuable. She wanted some real answers:

> The reason I did Computer Academy... because I thought it would open lots of doors, and kinda reveal more things, like seeing where else [computing] could be applied. Because right now, the only options I'm seeing are like, I don't know, working for Google, or some like very strictly computer science thing, but that's not why I did it...
>
> Google is the only [tech-related career] I can think of right now, on the spot. If you asked me the question, that would be it. But [I would like] to know more about *how* (emphasis mine) what I'm actually learning can be useful to so many different things, because I would like to know. Especially with college coming up.

If computing was so relevant to all sorts of different careers and opportunities, why couldn't she name any of them? Indeed, many classes of technologies (e.g. data science and analytics, AI, security and networking) and domains (scientific exploration, journalism, finance, the humanities) were conspicuously absent from students' imaginaries – perhaps because students had not yet been exposed to these alternatives. This of course had an impact on the potential futures that students could envision for themselves in relation to the practice of computing.

As a cohort, students named very few images of possible career trajectories (e.g. making games or working for Google) with the computing field. Moreover, these trajectories simply did not align for students who had strong identifications with other domains (environmental science, biology, medicine, sociology, fashion, etc.). Without being exposed to the many applications of computing, it was difficult for certain students to construct a future vision of themselves as computing practitioners, and hence form a sense of belonging within the practice

7.2.2 Reflections on Past Engagements

Engagement is what people do within a community of practice to keep it going, which involves doing work that is valuable to the practice (Wenger, 1998). Engagement can take many forms within the practice; not everyone does the same thing, nor would it work if everyone did. For instance, to make a software product, designers, engineers, marketers, sales, and operations people have to work together to accomplish different things that contribute to the whole. As Wenger describes:

> Unlike a classroom, where everyone is learning the same thing, participants in a community of practice contribute in a variety of interdependent ways that become material for building an identity. What they learn is what allows them to contribute to the enterprise of the community and to engage with others around that enterprise....We are all engaged in the pursuit of a socially meaningful enterprise, and our learning is in the service of that engagement. Our communities of practice then become resources for organizing our

> learning as well as contexts in which to manifest our learning through an identity of participation. What is crucial about this kind of engagement as an educational experience is that identity and learning serve each other (Wenger, 1998, p. 271).

Ideally, then, educational spaces should be designed so that there are multiple ways to contribute to a particular objective, so that not everyone is required to do exactly the same thing. Not only is this way of organizing work more true to practice, but it also offers more ways of participating, and thus more "material for building an identity." Hence, a learning environment should ideally support a variety of tasks and roles, which can be designed ahead of time, negotiated in situ, or usually a bit of each.

What makes a computing-related learning environment engaging?

At the beginning of the internship, students discussed what they hoped their engagement within the internship would look like, often in relation to their own histories of participation in computing. Simply put, students wanted the *process* of learning to be accessible and meaningful, and to explore interesting applications of computing while at the same time building competence. Two important criteria for a meaningful engagement were the ability to take initiative and be personally invested in an idea. As Angelo and Jose discussed:

> **Jose**: I'm actually with the [*Camino Contigo*[40]] group (see 4.3). We started a project like a year ago about social justice and how us like Latinos should be up to date on the Dream Act and DACA and all that. And, you know, how we should get together and fight against it using technology.
> **Angelo**: We were working, like, at our school (within the Computer Academy) we were working on a little bit of a [social justice] project, but like I said I don't know…it was kind of just, "you have to do this" so I'm not entirely sure where we want to go from there....
> **Jose**: On the project you guys were doing, did you guys see yourself taking initiative? Were you feeling passionate about what you guys were doing?
> **Angelo**: Not really. It's just kinda like {{inaudible}}.
> **Jose**: If you're going to do something, you might as well do something, like, might as well not…laughs.
> **Angelo**: Yeah. There's no real point.

Social dynamics were also important. For example, Jocelyn and Derrick mentioned the importance of camaraderie and/or "finding a good table" of classmates with whom to chat, get help, joke around, and ask questions. A good table could push you to do your best. A bad one could leave you feeling isolated and uncomfortable.

---

[40] The 'social justice' app that some of the students from Code 510 were designing and building.

Developing confidence within the practice, and having a clear path to develop it, was also a theme that was expressed over and over again in students' post-internship interviews. Despite having a year-long computer science course, most students did not believe that they had a command of "the basics" – a term that was never quite defined, but seemed to refer to a level of competence that was just beyond their grasp.

Two issues represented chronic challenges to their ability to learn the basics: (1) significant variability in prior knowledge among students, and (2) assumptions made by the teacher regarding how much prior knowledge students already had. This could be particularly discouraging to those who were new to computing. As Ada described it:

> My teacher would talk about it (some computer science idea) but he didn't explain it and so, like, someone like Nick is in our class and obviously he's been doing it for a while and like my partner that I worked with a lot, Damon, knew about all this stuff and I was like, um, "I'm supposed to be learning this stuff!? How come people know it and I don't?"And so, like, um, our first task when we got into class was take apart a computer and figure out all the (inaudible) and put the computer back together and, like, a bunch of people were like "This and this" (gesturing to people tinkering with the computer), and I was- me and Madeline were looking at each other like "We don't know what any of these things are. How are we supposed to do this?"

Why did Nick and Damon *already* know (or appeared to know) "about all this stuff," while Madeline and Ada didn't know what "any of [those] things [were]"? It seemed as if you *already* had to know a lot about computing to be able to learn more about computing, even in a supposedly introductory class.

Madeline expressed a very similar sentiment regarding prior knowledge:

> Like, if people come into your class and they don't even know what code is, talk to them as if they're babies, because honestly, they don't know anything about it. And I wished people did that to me, because I came in and they were just throwing things at me, and I was like, "I have no idea what's going on right now."

I examined the other post-internship transcripts to take stock of the variability of students' experiences with computing. Indeed, it did appear as if several of the male students had been engaging in software- and hardware-related tinkering long before high school. For instance, Nick, who was featured in Ada's story, shared that he had been programming for 3 years. As he described his experience:

> When I was really young I liked robots, and I wanted to make a robot, and then my dad found like a C, like programming in C book, and yeah, so. He gave me a programming in C book and was like, you want to make robots, you've gotta, like, read this....I just love making calculators in C. You know,

> 1 + 1. So I kinda did that for a while, and then I started delving into Java, and from there into Objective C and that kind of stuff. And then I got here.

Unlike Ada and Madeline, who were new to programming, Nick had experience in at least three different programming languages, some hardware experience (robots), and regular mentoring and guidance from his father.

Other students also had substantial prior knowledge of programming, even without help from parents. For instance, Erick learned some coding while participating in various online communities, and hanging out with his cousins:

> **Erick**: Um, well, kind of I've always been interested in computers...like I remember when I was like 5, going to my cousins' house…I have a lot of cousins, and like they started getting computers, and I was like, what's that? And then I just started playing around with things, and I wanted to learn how to do stuff, so I started looking up how to do stuff.
> **Sarah**: Interesting. So do you remember some of the first things that you wanted to do on the computer?
> **Erick**: Well, I started with paint, and like drawing things. And then I moved onto sort of, like, coding kinds of things and graphics, stuff like that.
> **Sarah**: So your first experience with coding, was that on your own, or like in a class?
> **Erick**: On my own. I used to play this game, and there was a community building up around this game sort of thing. And people started doing their own blogs and stuff like that. And so I got into that, and like making sure…like getting a blog that looked the way I wanted to. That was when I started using code and actually learning what different things mean.
> **Sarah**: And these were things you just learned on your own? Did you ask other people?
> **Erick**: No I don't really have anyone who really knows anything about that.
> **Sarah**: So you've kind of been teaching yourself things for a…you've had a long history of doing that?
> **Erick**: I mean I've been learning actively, but I have my hiatuses too…in between, where I kinda forget some stuff.

Whereas Nick had been mentored by his father, Erick's trajectory of engagement was more informal, moving from playing with his cousins, to participating in an online community organized around an online game, to creating a blog, to finding online resources, to help him get his blog to look the way he wanted to (this kind of organic process of coding has also been documented by other researchers, e.g., Perkel, 2006).

Madeline, Ada, and others also had significant computing-related knowledge – It just looked different. Almost all of the students reported using Instagram, Snapchat, YouTube, and/or Spotify daily, and therefore were regularly searching, browsing, filtering, commenting, sharing, and so forth across a variety of platforms. With these experiences came particular aesthetic sensibilities, a situated understanding of the social media landscape, a sense of what features might be useful, frustrating, missing, etc., a sense of how content traveled, and some understanding of information retrieval including the relevance, reliability, and accuracy of content.

Perhaps it wasn't just that some students' histories of experience were more relevant to computing and others' experiences were more peripheral. Maybe certain computing practices and roles were still being privileged within the introductory computer science classroom. In particular, there were not clear avenues for students with particular histories of experience to connect their prior knowledge to the activities in computer science class. Whereas deconstructing and putting a computer back together would certainly make more sense to a student who had experience building robots, students' facility with social media use did not have a clear mapping to many of the initial topics that are typically privileged in an introductory computing course.

7.2.3 Challenges in Finding Alignment

A final mode of belonging, according to Wenger, is alignment. Alignment links engagement to imagination by allowing a person to understand *how*, specifically, one's engagement is relevant to one's image of the practice (and other practices). To exemplify alignment, consider the following scene from the '80s movie, "The Karate Kid," that happens just after Mr. Miyagi (a Karate master) agrees to train Daniel (a high school student who is being severely bullied) in Karate:

> During his first week of "Karate training," Miyagi makes Daniel wax cars, paint fences, sand floors, and so forth, for hours at a time. While compliant at first, as the week wears on Daniel grows increasingly frustrated and confused, believing that Miyagi has played him for a fool – using him for free labor. Just as Daniel is about to storm out of Miyagi's workshop for good, Miyagi stops him and tells him to perform the motions he has been doing all week. "Sand the floor" he instructs (i.e. move your arms in big circles). "Wax on. Wax off" (i.e. move your arms like you were when you were waxing my car). As Daniel complies, Miyagi attempts to strike Daniel. However, the movements that Daniel had been "practicing" through his labors block each of Miyagi's blows. "Come back tomorrow," Miyagi says to Daniel, who is clearly bewildered by his newfound abilities.

The "wax on wax off" scene exemplifies what it means for engagement to come into alignment with imagination. Although Daniel had been *engaging* in the practice (doing chores) and *imagining* a future where he commands respect as an excellent fighter, he had no notion of how doing chores and developing Karate expertise were linked. It is only when Miyagi physically shows Daniel how "wax on wax off" relates to the fundamental defensive blocks of Karate that Daniel's practice comes into alignment. After seeing the value of the chores he has been doing, Daniel became motivated to continue training, and ultimately goes on to win the Karate tournament.

Within formal spaces of education, Wenger argues that issues of alignment come up all of the time, particularly when connections between "real world" practice and classroom practice are not made visible, or when they don't align in the first place.

Within computing education, questions of alignment might include: how does making a cartoon in Scratch (a blocks-based language) relate to what Google or Apple do? Or, how does learning linear algebra relate to being a video game developer? Learners need to connect the means (i.e. current engagements) and ends (i.e. aspirational imaginaries) within some coherent framework. Otherwise, learners can have difficulty identifying with the immediate practice, even if the practice is very relevant to the student's broader enterprise or goal.

During the internship, students reflected on two different alignment challenges. The first related to students' ability to see alignment between the practices in which they were engaging in computer science class, and broader computing practice outside of the classroom. Consider the following conversation between Angelo and Ada:

> **Ada**: [Teacher's name] didn't teach us much.
> **Angelo**: Yeah, we learned nothing in that class at all.
> **Ada**: Yeah, I want to learn as much as I can 'cause I didn't learn anything in his class.
> **Angelo**: Yeah, exactly. Just make up for that wasted year in [teacher's name's] class…and actually do something.
> **Ada**: Yeah, learn a lot of stuff, or like, learn stuff that I could do…like if he gives us an assignment, I actually know how to do it [instead of being], like, so confused.
> **Angelo**: Yeah, I know on that last assignment we just Googled how to do the entire thing instead of using anything that we learned – which was, like, nothing.
> **Ada**: Yeah, like everything that you do, in all the videos, like you just look at the tutorial because like he…
> **Angelo**: Like we taught ourselves in that class.
> **Ada**: Yeah.

It is clear that both students had a sense that they had been left to their own devices to figure things out. This involved Googling "the entire [assignment]," watching videos, doing tutorials, "teaching oneself," and/or sometimes experiencing a prolonged sense of "not knowing" that never resolved. While working through these different situations are certainly part of what it means to be competent in "real world" computing-related endeavors, for Angelo and Ada, these experiences were not sufficient to create a sense of alignment with authentic computing practice. While Angelo and Ada believed *that* computing was valuable, they didn't understand *how* their learning practices related to the broader world of computing.

Whereas Ada and Angelo's critiques related to what constituted valid participation, a second issue revolved around understanding the relevance of the activities themselves, and how they connected to the broader landscape of computing:

> Madeline: At school, I really didn't find myself connecting to code. I didn't...maybe because I wasn't exposed to it correctly...they were just like, "here it is." And I was kinda like "What?" And they never really went into detail and got me thinking about *how*.
>
> I think you need to give people a chance to understand it, even if they won't like it. I feel like you have to give 'em a fair explanation of what they're getting into and then they can choose if they like it or not. And so the reason I want to do [Computer Academy] next year, and I'm still staying in it is because, like, I wanted to learn more because I didn't learn what I thought I was going to learn the first year. So the only reason I did it (continue for a second year) was out of curiosity. Because I want to be taught more, I guess.

Madeline was seeking out "a fair explanation" of *how* coding related to the broader computational world around her. She wanted a baseline understanding of computing technology and infrastructure, even if she ultimately decided not to pursue it as a career path. Her curiosity continued to drive her participation in the computer academy, despite the fact that she hadn't yet connected to the practice.

7.3 Engagement within the Internship: Connecting Identities to New Practices

I now turn to students' engagements with the internship activities themselves. As described in the "Course Summary" section (see 5.4), students spent the last week of the internship designing and building their own third-party web apps, which incorporated data from various networked platforms (Instagram, SoundCloud, and Twitter). To facilitate this process, I gave students some "starter files" that included a simple template (pictured in Figure 22), and a JavaScript file with some code for hooking into each provider's API.

I based this template on Gabriel's idea (recall: "I would like to create an app that gets Instagram, Facebook, and oh, Snapchat all together into one…"). It was each student's job to modify this code as they liked, or to start from scratch if they preferred, in order to create a web app that was interesting to them. Students were able to pursue any aesthetic, any theme, and choose from a wide variety of media – both in form (music, images, videos, text) and content (any topic available on the internet).

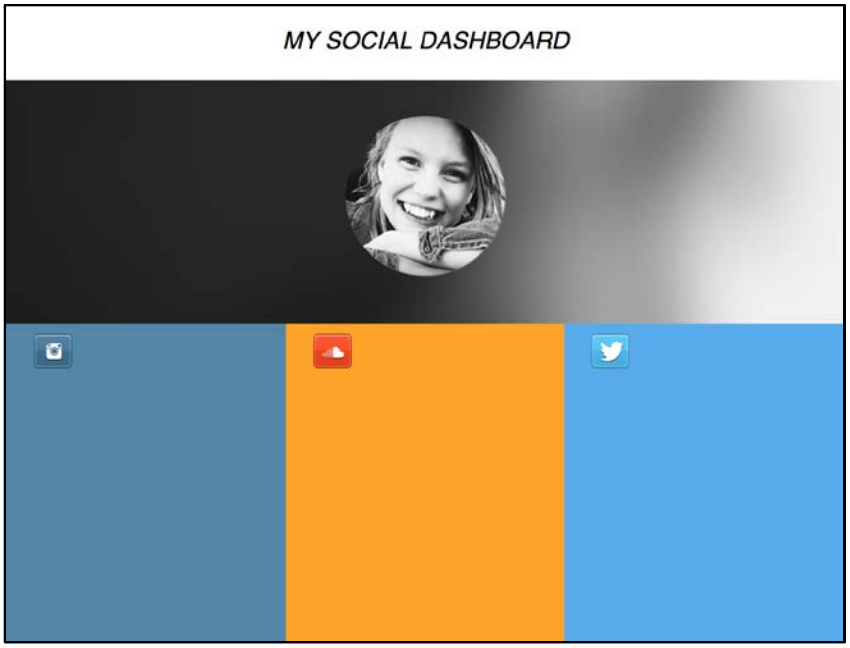


Figure 22: Initial template (with social media hooks)

Using transcripts of students' final presentations and screenshots of their respective projects, I discuss how students appropriated web design, programming, and networked data to share aspects of themselves and their identities. I have grouped students' interests into three categories: music, social issues, and utilitarian to highlight how old and new identities came together through the medium of web development.

7.3.1 Music

I begin by focusing on one particular student, Amar, and his love of music. Amar expressed a strong interest in music throughout the internship. As the youngest intern, he had just completed his freshman year at Realm, and was the only student in the internship who did not know anyone else coming in. During the icebreaker activity during the first week (where students were asked to describe some of the apps they regularly used and what they wanted to learn during the internship), Amar used his love of music to find common ground among his classmates:

> **Amar**: Hey, My name's Amar. Uh, So apps…I like Spotify, because I love music, um, I love Twitter because it's fun to be on there. Um, yeah, that's all I got. I don't have any app ideas.
> **Jose**: I'm Jose, I like YouTube, Steam. I also like music. It's chill...
> **Amar**: What kind of music do you listen to?
> **Jose**: Anything, dude. I'm open to anything.
> **Amar**: I listen to rap, but I'm mostly open to anything too.
> **Jose**: Rap is- (pauses). I kinda like, you know {{ artist name}}?
> **Amar**: No, I've never-

> **Jose**: You should listen to them. I mean they talk about some deep ass shit (laughs).
> **Amar**: Yeah, yeah.
> **Jose**: You should listen to them. I mean, at first the content is, like, really heavy, but I really recommend.
> **Amar**: Yeah, yeah, yeah I like that deep ass shit too.

Within two minutes, Amar made a connection with Jose through music, as both students revealed that they enjoyed rap that involved "some deep ass shit." Music was a way for Amar to reveal a unique aspect about himself and signal his tastes and interests, perhaps in a way he believed that might resonate with Jose. Amar's interest in music also showed up in the app that he made. As he described in his final presentation:

> So ah, my idea was, I wanted a place where you could access these music artists' social medias [sic] easily, so that way you can connect with them. Because, you know, music is a passion of mine, you know. I grew up playing guitar, and so…and then you know, listening to rap or whatnot. You know, [it] really changed my life, and you know, it was a struggle just to learn about the artists. You had to do it on all of these websites, you had to go on, you know, like YouTube, and you wanted to connect with them, but you had to go on like Twitter and you had to, like, do so much. I wanted one place where you could just go and, like, see what they're talking about. See what people are talking about, like why they're talkin' about 'em. So I created this app.

Amar chose to talk about his reasons for making the app in a way that was very personal: music was a passion for him, something that "changed [his] life," and pursuing this passion had been made possible, in part, by utilizing YouTube and Twitter. He understood, first hand, what this process of learning about and discovering new music looked like and how meaningful it could be, and therefore wanted to create something that allowed others to experience it as well. Amar continued his presentation by describing how his app might accomplish this goal:

> So let's say a friend told you to listen to someone, and you know, you're like "Oh, OK." And so you go on this website and you search their name, and the name comes up along with their Twitter, their social media, and their SoundCloud if they have a SoundCloud. I was originally going to use Spotify, but I didn't have the time. I mean I did {{inaudible}} but stuff happened.
>
> Um, the technologies I used were SoundCloud, Twitter, and Instagram. And I just…the way that it works is like I use these APIs by searching tags…it doesn't matter the name, so the one that I did was like "Lana Del Rey." And what I did was I just searched up Lana Del Rey, and you know, and when I go on the website, I'll show you. There's like people talking about her, and using her hashtag and talking about her under her name (live demo of his app displaying information about Lana Del Rey).

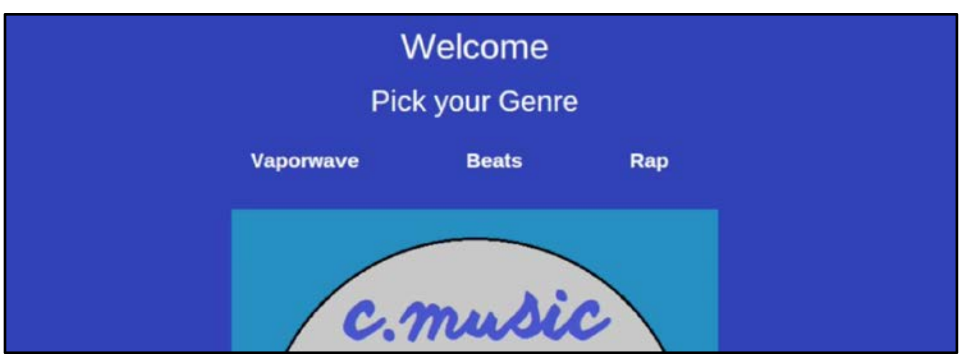
Amar demonstrated, through his live demo, precisely how this process of discovery would work: how you could search someone's name that you didn't know, say, Lana Del Rey, see what people are saying about her, and listen to some of her music. In essence, Amar had imbued his own process of discovery into the app that he built, so that others could follow that same path. Finally, Amar connected the app that he made to the "person [he] is" – to his identity:

> And you know, I want people to discover new artists, because I feel like new artists…that's how I, you know, am. The person I am. Because I discovered these new artists....I also want to add like something where it tells their name and a little bit about themselves. Like a little bio so you know where they come from. So you know. If they come from, like, the hood or something, you know, let's say you're like born in the hood, and you can connect with them. You know, you know how they feel, their struggles. So.

Amar's app was not only about discovery, but about connection – feeling the struggles of an artist, and then relating them to your own struggles. Appreciating music was not just about the music itself, but also understanding the background and context that went into producing music. Twitter, a dynamic, networked media source, enabled a potential user to connect this context to the music pulled from SoundCloud (though he wished it could have been Spotify).

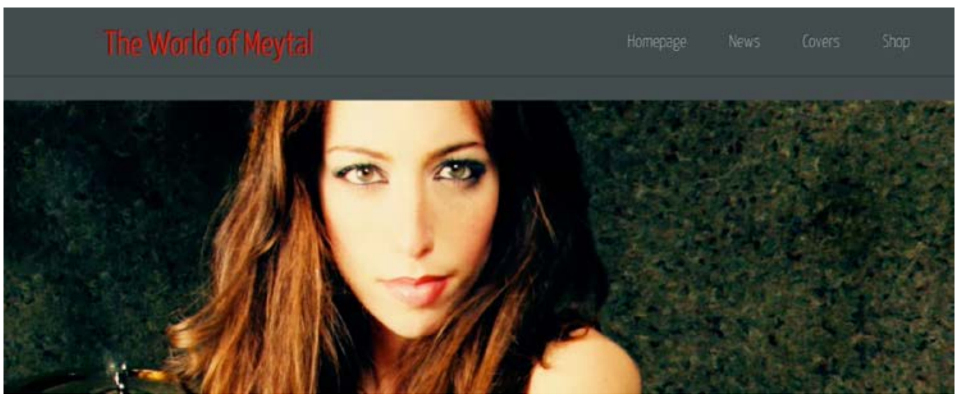
Sasha pursued a similar trajectory in his app, which was also intended to help people broaden their musical horizons by sharing his own expertise of more "underground" artists. To do this, he created a browseable music interface that issued queries for *particular* artists (hardcoded into his app) to SoundCloud. After retrieving the data, his interface then generated playlists and an audio player on-the-fly. As he explained:

> I wanted to open other people to the world of music. Underground music you don't hear on the radio. Because there's a lot of things you can do with sound, believe it or not. A lot of people just listen to what's on the radio, and you know, that's cool and all, but there's other stuff. There's other stuff. And it might make you happy if you listen to other stuff.



Two other students chose to feature one artist exclusively, also citing a very personal connection to the artist and/or a particular genre of music. For instance, Derrick chose to feature one of his favorite artists, an Israeli heavy metal drummer:

> For my project, I decided to do a fan page for my favorite drummer on YouTube. She's really amazing, she um, she's like any one of us…she grew up, high school and stuff (pause). I find most of my music through her [YouTube] channel. There's a lot. So my project was designed to direct them to specifically her, so they can like, connect to her and also like, be able to like, find common ground with her so that, um, they can find interest into the music. Um. but I just…with this I just really wanted more people to enjoy the music than her, really. I just wanted…while the attention may be on her, but it's, it's more about the music, that's what my vision was for.



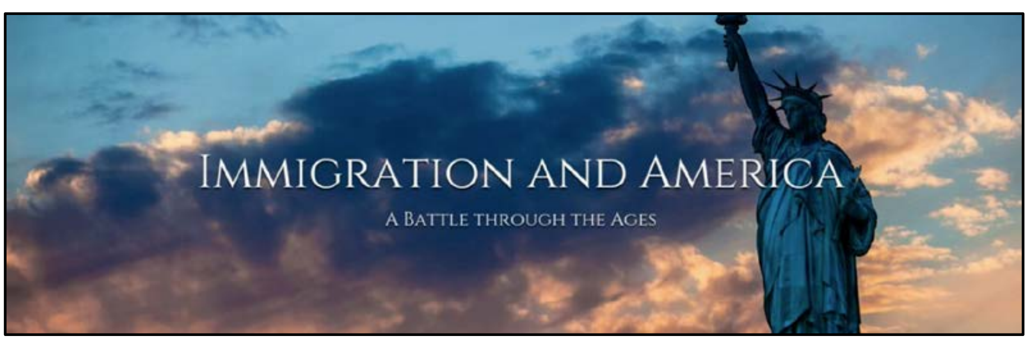
Ling also created a fanpage, dedicated to her favorite Korean boy band, BTS. Her presentation, included whimsical animated gifs of cartoon cats and a member of the band being dragged into a pool:

> The main point is, I guess to promote the band, and what's unique about it is that it lets fans sort of know different information about the group in one source, I guess, in this one site, "cause for me, I'm a fan of them, and usually I have to go to different medias to look for different information about them. So like if I want to hear their songs, I have to go to SoundCloud, and if I want to see, like, pictures and stuff, I have to go to Twitter, so it's like all over the place, so I decided to make all of that together.

7.3.2 Social Impact

Two apps, Erick's and Anna's, focused on issues specifically related to immigration. Like the music apps featured above, these apps also functioned as representations of students' knowledge, interests, values, and identities. Erick, Anna, and others had been observing a spike of anti-immigrant sentiment throughout the internship. This broader cultural discussion, which had flooded their social media feeds, had been fueled in part by Donald Trump's anti-immigrant and anti-Mexican rhetoric on Twitter (this was over a year before he was elected president). As Erick explained to his classmates:

> The purpose of this app was to, um, bring together and paint a better picture of the current conversation that America is having as a whole about the immigration situation that's currently happening. And my goal is to, like, make sure people become aware. Those people who are being affected by this and people who want to learn more they become aware of how it's been developing throughout the years, and how it's taking (pause) how the conversation is taking shape through um, social media. It's becoming a big thing in our news and stuff.



Unlike Amar, Derrick, and others, Erick did not explicitly mention the personal nature of his app nor take a position on it, opting instead to frame his app as highlighting a "conversation" that was "taking shape" through social media. Erick then transitioned into a description of his interface – one of the most sophisticated and technically complex in the class:

> The interface is kind of good (laughs). And appealing. I wanted to have, like, an immersive interface, so it's like a Parallax effect. And due to that, I had to have a lot of complications and tried to work around that. And it works by using APIs. It draws from Instagram and Twitter currently only. And it uses the basic things like HTML, CSS for styling, JavaScript to do all the APIs. And the goal, drawing from those, Instagram and Twitter APIs, is to paint a picture with words, as well as with pictures.

Erick made it clear that he was invested in the aesthetic, interactive, and technical aspects of his app, and was happy with how it turned out, describing it as "good" and "appealing." He also described his app as artistic, intended to "paint a picture with words, as well as with pictures" – to convey information, but also a feeling / experience of the immigration debate.

To achieve this goal, he chose to completely abandon the starter template and introduce significant complexity into his code in order to pursue a "Parallax" effect – a kind of popular scrolling animation that works to animate images and text at different speeds. The decision to scrap the starter template to pursue a particular aesthetic was common. In roughly two-thirds of the projects, I noticed students going to great lengths to figure out how to implement particular designs, sometimes spending hours positioning an image a particular way or getting an animated image carousel to work. Erick continued:

> I'm targeting people who want to learn more about immigration – the conversation on immigration, and I also want to put out resources to people that might not know they're there to help them with their immigration if they're affected by it. And, yeah (pulling it up / live demo).
> This is…scroll down, and stuff, there's like slides, and that's like the parallax effect. ...Right now currently it just shows your timeline, and I'm aiming to put in certain hashtags and stuff. And like the Department of Homeland Security has its own Twitter, so I'd link to that also, as well as activist groups who have like Twitter groups. I'll link to them too and stuff. And Instagram is right there...it should as well target those hashtags and groups. Like, the group that has the immigration thing inside of Instagram itself kind of thing. So link to them too for information.
>
> The way I'd categorize it would be anything immigration related, like maybe Homeland Security regarding the yearly annual report on how many deportations happen and how many people they detain entering the country, as well as unofficial sources, so say like schools who take surveys. How many of their students are undocumented sort of thing. And yeah.

Erick described how the app worked with the Twitter and Instagram APIs, including the hashtags and accounts he would target, and the kinds of information and resources he hoped to provide his potential users. In other words, Erick was curating his own knowledge of various existing initiatives and groups that he already knew about, including Homeland Security's deportation and immigration statistics, and various activist groups who were already organized on the platforms.

Like Erick, Anna also created an app that featured immigration-related news. As mentioned in Chapter 5, it was Anna's commitment to immigration activism via social media that had inspired the design of this internship in the first place. As she explained during her presentation:

> Most of the news that people see are like news that affect everybody, or they only cover some things, or they only highlight certain things, so my app is a news-based app using hashtags to collect the data, and it was mostly centered towards immigration, um, but there are like parts of the news in there...I'm using right now Twitter and Instagram and the hashtags to collect, um, posts that people have been putting up about immigration, and, um DACA....there is like charts of like, percentages of different things, and there are like posts about Obama, and things like that.
>
> And…let's see Twitter…here are some posts about Twitter. There are not that clear right now. These are just posts that people have put up recently, so I'm going to try and filter it to certain, um, accounts that put up the news that are more reliable.

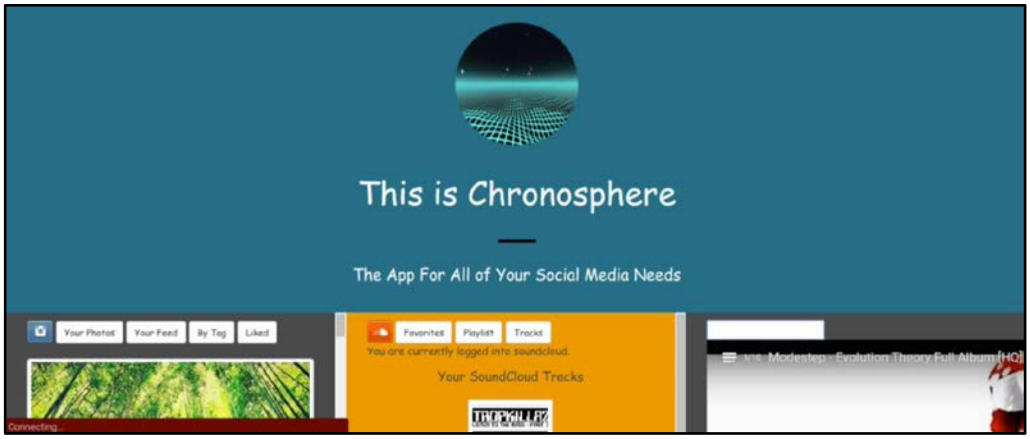
By creating issue-focused apps, Anna and Erick participated in the same way as the "music app" students had, curating a pathway through a particular set of resources, ideas, and discussions, and sharing a side of themselves that they wanted to highlight or make visible.

7.3.3 Exploring Creative Expression on the Web

One final genre of app, pursued by a little under half of the students, did not center any particular interest or topic, but explored the more functional aspects of web design and web programming. Teddy's app, for instance, was a way to flexibly and conveniently display any content, in any way that a user wanted:

> So my app is called Chronosphere. I combined Instagram, SoundCloud, and YouTube. And basically that's it. But I wanted it to be completely customizable for people, so you could have it in any order you want and in any colors [sic] you want, as well as setting your profile picture kind of like the other sites.
>
> Basically I just want the SoundCloud to play whatever people want…what the person wants to listen to, and I want the Instagram feed to refresh just when they click on their feed button. So like, and also I want them to be able to search their Instagrams and have other pictures come up. And basically use the APIs to pull from whatever accounts you're logged into. And the

> YouTube player can go fullscreen, which is kinda nice. And…I just…again used the APIs.



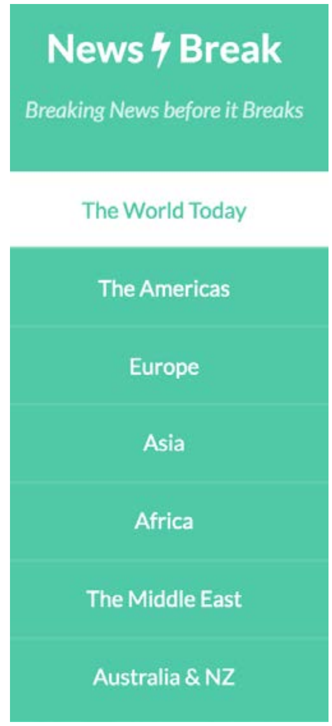
> Basically (pulls up the code), this is most of the JavaScript I think. I don't know. I copied one of them. But, I had to get…from the APIs, I had to get all of the data to my site, and get it to log in and have all of the access tokens and that was kind of a pain.

Teddy's innovation was the way in which he enabled potential users to flexibly query and style various forms of content together: the audio player, photos, and videos could be re-generated on-the-fly, based on user preferences. During Teddy's presentation, he also pulled up and displayed his source code on the projector, pointing out the places that were particularly difficult or novel, including code he had written (with some help from me) to query and display custom videos from YouTube (an API that he had to figure out for himself, as there were no existing examples provided). YouTube functionality was another way in which his app was unique, allowing him to showcase his emerging technical skills, which would be of potential interest to others. Teddy also gestured to the way in which his creation might potentially get used more broadly: "And I really want to put this to full use and just develop it all the way so that it can actually be something that people will use."

Four other students also took this approach, focusing on functional aspects (e.g. how to incorporate widgets, change fonts, icons, and colors, how to make an animated photo carousel, etc.) over a particular theme or idea. For instance, Angelo, who did not appear particularly interested in the content on Instagram, SoundCloud, or Twitter, chose to spend his time completely rebuilding his own social media dashboard from scratch. He chose a "Galaxy" theme, and integrated an animated menu widget that he had found on Reddit. Instead of focusing on any particular message or representation of self, he focused on aspects of web programming that he was curious about.

Nick – who took on the role of the class clown throughout the internship – created a "breaking news" app (i.e dashboard) that explored how he could filter news by geographic region. Categorization became problematic for this project, in particular whether it made sense to group Australia and New Zealand. Consider the following exchange during the Q&A session after Nick's presentation:

> **Ling**: Why do you hate Australia?
>
> **Erick**: No, no, no, no. Let me handle this (laughter). OK, Australia should have its own category! Not with New Zealand...Equality for Australia!
>
> **Nick**: Let's be honest. What happens in Australia? Koalas attacking polar bears and other penguins?
>
> **Erick**: You can raise your children with morals in Australia! (laughter). So, your complaint box does not work right now... (laughter)
>
> **Nick**: It especially doesn't work if you sign in with a username that has Erick in its name. (smiling)
>
> **Erick**: Will it ever work? (laughter)
>
> **Erick**: Answer my question! Will it ever work!? (smiling)
>
> **Nick**: It will work. I'll add the mailto: thing and then you'll be able to mail your stuff to me and then I'll put it in the junk mailbox and I'll look at it in a few years. (laughter).



While the debate is playful and not particularly consequential to students' core beliefs and identities, it suggests that decisions around how an app separates, filters, and categorizes information can become a generative way to help students examine how and why particular design decisions are made, and their potential consequences.

7.4 Mediation: Materials for Building Identity

In the findings above, I describe students' goals and motivations at two time scales: one that encompassed their broader educational trajectories (i.e. why they came to the internship and what they wanted from it); and one that was emergent and local (i.e. contingent on the tools and resources available, and the people who surrounded them). In this section, I consider how the broader contexts of learning mediated these various experiences (RQ3).

Regarding their broader reflections and visions, shaped by a variety of past experiences, students wanted to make "something" that was fun, simple, beautiful, useful to others, and exciting; while at the same time developing a particular kind of computational literacy – a "basic knowledge" of "what you use everyday" – that could ideally help them complete their assignments in computer science class.

Students also strongly believed that learning about computing would help them secure a favorable position in a computer-mediated future, but weren't entirely clear on *how* computing knowledge figured into this future, other than potentially enabling them to work at Google (an idea that was met with mixed reception). Students wanted "the basics" and "the big picture" at the same time, while pursuing goals that resonated with their values. From this perspective, designing a "situated" learning environment for young people was not as simple as engineering an "interest-driven" (Ito et al., 2010) learning activity, or "building a concrete, shareable artifact" (Papert, 1980). It was also about alignment: helping students to (a) connect the practices they were learning to their broader, technologically complex world, and (b) figure out how they might want to participate in this world going forward.

Within the internship, I found that students appeared to be driven by their desire to achieve various functional or aesthetic features within their website; share unique aspects of themselves; and/or make each other laugh. The artifacts that students built offered one window into these motivations, given the content, features, and aesthetic elements that students selected and/or implemented. Students' presentations, discussions, and reflections provide another into *why* particular features were pursued, and what these meant to students.

While the activities were designed to be interesting to the students, it was up to them to make them meaningful, given their particular goals and motivations. Therefore, in the rest of this section, I address RQ3, by considering how the opportunities and constraints afforded by the final project activity mediated students' participation. I do this by focusing on three key resources that shaped students' engagement in the projects: (1) content, (2) communities, and (3) design opportunities.

7.4.1 Content

All of the students participating in the internship reported being regular users of multiple social media platforms: YouTube was universal; Facebook, SnapChat, and Instagram came next; and a long tail of other platforms followed (e.g. Steam, Spotify, Tumblr, DeviantArt, and so forth). The web offered the students a near endless selection of dynamic, queryable, and ever-changing content (e.g. music, gossip, political issues, Australia) in a variety of formats (images, videos, audio files, charts, raw data, text, etc.). Even if the focal social media platforms were not ones that every student regularly accessed, each student was familiar with the concept of a platform, and could reasonably expect to find some connection to a theme, idea, or interest.

Given this wealth of prior experience, over half of the interns pursued an app relating to a particular interest or issue by tapping into social media content that they regularly accessed anyway. Amar and Erick did this by reconstructing and streamlining a particular process for learning about music and immigration respectively, based on their own media practices. Derrick, Youngshi, and Sasha's approach involved carefully curating specific artists, images, and music to fit a particular set of criteria, drawing from their own expertise. These projects enabled the student to present and communicate a passion or interest, based on content produced by other people on the Internet, and allowed others to explore, learn, and "connect" to people and ideas relevant to this interest.

7.4.2 Audience and Community

Another affordance of working with social media platforms was that the content was never separate from the people who produced it. As a result, students were able to connect their interests, values, and ideas with a broader network of people (this has also been noted within the Scratch online community, e.g., Dasgupta et al., 2016). They could actually demonstrate how their own passions and ideas were part of something larger – something that was unique, but also endorsed by others, and always dynamic and alive. As Derrick expressed when reflecting on his "The World of Meytal" app:

> I liked being able to, like, use like the entire Internet. Like the entire Internet is at my disposal to like reuse, and for other people to be able to view it conveniently

Similarly, Erick and Anna, through their respective immigration apps, were channeling young Latino/a activists and artists who inspired them as they described their own passions and interests. At the same time, they were also enabling their potential users to get news and events pertaining to their local area, and to get ideas regarding how they might like to engage with the issue. Similarly, Amar was enabling potential users to "change their lives" by connecting to music and artists, just as he had. Each app creator effectively configured a particular pathway through a disparate group of online conversations and resources associated with their topic of interest in a way that reflected some version of their own process of learning and participation.

Even for students who didn't center any one topic or community in their app, the idea that other people would be using it to connect to people and ideas was still very much present. That their apps would be seen by a worldwide audience was very motivating. As Teddy expressed, during his post-internship interview:

> I think just seeing something that you made be used and, like, liked by other people just kind of fun. It's one of your creations is out in the world now people are using it.

7.4.3 Design

Design allowed students to present content in very particular ways. Some designs were purely functional: for instance, several students architected their app to only allow access to a finite set of links, tags, and search terms to create specific paths through the content, while others allowed a user to search across several social media platforms at once using any keyword. Other designs were more directed towards look-and-feel: attending to fonts, colors, graphics, logos, icons, layouts, and various animation effects. At least four students, when asked about the most challenging part of their project during the final presentation, responded with an anecdote about implementing a design feature. Two reported that they spent "literally hours" getting their photo gallery to successfully animate, or making their dynamic menu to show and hide when clicked, but were glad that they had, and happy with how it turned out.

In short, aesthetic and design goals were a key driver of sustained engagement, even in cases where students did not express an interest in social media or in a particular topic area. Whereas students could have easily kept the starter templates and directed their efforts towards content-related goals (which would have been significantly easier), for many of them, the time investment in design was worth the effort.

7.5 Conclusion

The purpose of this chapter has been to consider some of the implicit and explicit drivers of meaning and belonging within computing – both within and beyond the internship. Through students' voices and artifacts we see that students are curious about computing as both a way to design and build interesting artifacts, as a vehicle for opportunity (college applications and career-related skills), and as a cultural phenomenon that seemed inextricably (and sometimes inexplicably) tied to their future trajectories. Given the importance of the web as a site of cultural production, I argue that building a web app offered young people new ways to participate in their computationally-mediated world.

8 CONCLUSION AND IMPLICATIONS

The research described in this dissertation began with the assumption that expanding access to computing and data education could be an important lever to promote social equity. *CSforAll* would allow people who were categorically underrepresented in computing to finally build and shape computer-mediated systems – ensuring that their interests, priorities, and cultural stories were represented and accounted for within computing technologies. Given that computer-mediated systems are embedded in so many aspects of our lives, shaping what we can and cannot do (Lessig, 2000), software designed by more diverse teams could translate into better and more equitable sociotechnical systems overall. And of course, underrepresented groups would finally have a seat at the table within the Googles, Microsofts, and Apples of the world, or any number of other high paying, high-opportunity technology industry jobs. In short, by ameliorating inequities in access to computing/data education, it would also be possible to address, at least in part, broader inequities.

My task therefore became designing tools, activities, and pedagogical supports that would be exciting and empowering for young people from non-dominant social groups (i.e. the majority of students attending public high schools in the East Bay). Perhaps by showing students how powerful data and computing could be – for lifting up their voices and stories, naming the world for themselves, and solving problems that were directly impacting their lives (Freire, 1996) – I could address the oft cited critique that computing education is not culturally relevant for many students (C. H. Lee, 2012; e.g., Ryoo et al., 2013). My subsequent explorations into data-intensive advocacy (chapter 3) and social justice apps (chapter 4) were efforts in this direction.

These inquiries revealed that while computing and data were capable of many things – measuring and analyzing data about the social and natural world (e.g. the Air Quality project); documenting and sharing assets, challenges, and ideas (e.g. the Planning project); or helping people to access existing information and opportunities (e.g. Camino Contigo) – the entrenched social and environmental challenges that the students and I continued to encounter extended well beyond what these technologies were capable of solving, especially absent other supports. In short, both the data→knowledge→action (as critiqued in Fiore-Gartland & Neff, 2015) and "computational action" (as put forth in Tissenbaum et al., 2019) ideals fell short of their promise.

One implication of these studies was that perhaps what was really needed was a more realistic accounting of students' computationally mediated worlds (explored in chapters 5-7). The more I listened, the more I could hear students asking, in Madeline's words, for "a fair explanation" of computing and its applications (p. 109) – the good and the bad; the inspiring and the mundane. Moreover, this fair explanation involved participating in contexts that were sometimes playful and fun, sometimes tedious but useful, and sometimes explicitly connected with the broader sociotechnical systems that framed students' lives.

I am reminded of an interview with Ta-Nehisi Coates – a journalist and writer who focuses on the lives and histories of people of African descent – that I recently heard on NPR (Tippett, 2019). I the interview, various audience members ask him to weigh in on how to think about supporting students to navigate a world that it profoundly inequitable and unjust. In the excerpt I share below, Coates has just finished speaking about about culture, history, and white supremacy, and is answering audience questions, many of which are asked by teachers:

> **Tippett** (interviewer): So, this question is from North Shore Country Day School middle school teachers. So it's back to teachers.
>
> **Coates**: Uh oh.
>
> **Tippett**: How can we help our students remain optimistic under this administration...
>
> **Coates**: Oh my god are you serious?!!!! (laughs)
>
> **Tippett**: ...when we ourselves are struggling?
>
> (audience laughter)
>
> **Coates**: Why would I know that?!!! (laughter)
>
> Um, OK. I'm going to try...I have to try (laughs). Um. How do you help your kids...I would probably reject the premise of the question. If I were in your class and I put myself back there, I don't think, even at that age, that I was looking for hope from my teachers. I think I was looking for enlightenment from my teachers. I think I was looking for exposure. I think I wanted to see other things about the world. I think I wanted to be exposed to different worldviews.
>
> Um, I think I like [pauses]. If I were a kid right now, I guess I would wanna understand, like, why did they kill Eric Garner? Like why is that OK? And the answer doesn't have to be, you know, I don't need you to make me feel good about that. But I need to know what happened. I just need...people deeply underestimate the freedom that comes from at least understanding. You know, um. It's one thing to be terminally ill. That's bad enough. But to not understand what's happening to your body […].
>
> I probably would want to be pointed...not even would want the answers but give me the tools. Arm me. Allow me to be able to understand *WHY*. You know? That probably would be more important to me. That's not hope. That's not hope. But um. That's probably the perspective I would have come from at that age (Tippett, 2019, minute 65:29, unedited version).

As Coates (and many of the students featured in this dissertation) remind us, young people are already living in the world of today. There is racism and police violence. There is surveillance capitalism, fake news, and election interference. And there is figuring out who you are and what you want in a competitive world where constraints and opportunities are both unevenly distributed. Given these realities, perhaps what young people want and need is to be armed with the tools to reason about the *how* and the *why* of computing, to figure out what is hype and what is real. In short, to be enlightened. Of course this does not mean that students can't also be creative, to explore, and to dream (Coates is himself a writer of fiction, and a lover of comics, hip hop, and the arts). But dreaming can also profit from an understanding of the world as it is, and an awareness and an acknowledgement of real life constraints and opportunities.

8.1 On Becoming "Enlightened" About Sociotechnical Systems

So what does it mean to become enlightened about data and computing in this current historical moment? And what does it mean to give young people a "fair explanation" about the computer-mediated, sociotechnical systems that surround them? I don't have authoritative answers to these questions, but I do think that trying to answer them is a useful exercise. Therefore, to conclude, I reflect on what I've learned about what is important to know about data and computing, and how we might help give young people the fair explanation they are asking for.

8.1.1 Honoring the Context

One aspect of becoming enlightened would involve being able to reason about the strengths and limitations of the medium in real situations with real people striving to achieve real goals. Whereas *context* in computing and data education is often treated as the subtext to motivate particular skills and strategies, the "non-technical" goals and practices within a context are most often first-order pursuits in their own right, which must be understood in their own terms. Alice and Scratch are platforms for learning about programming while telling stories or making games. But storytelling and game design are each well-developed art forms with their own conventions, tools, skills, and strategies, which are also important. Similarly, participating in data-driven advocacy involves knowing *what advocacy really means* – understanding the politics of a community, the nature of the problem and its root causes, effective strategies and tactics, how to build coalitions, and so forth.

If learners are to develop a real sense of what computing is and is not capable of, we must offer them a fair treatment of the context itself. For example, if the the curriculum asserts that data visualization is a powerful journalistic tool, then students ought to be able to understand both what makes a journalistic piece good, and how data enhances some valued aspect of journalistic practice. If data is positioned as a powerful tool in the fight for climate justice, then students should have the opportunity to show a climate denier a line graph of CO_2 emissions and average temperatures over time, and see what happens. Does it make a difference? And if not, *why*?

These kinds of grounding criteria can help ensure that educational initiatives don't inadvertently exit reality and reproduce deterministic narratives about the power of computing and data, which can be more fiction than fact (Ames, 2019; Sims, 2017). I argue that by offering a richer treatment of context, learners can be supported in taking a more holistic approach to participation and problem solving, where computing and data are treated as one means (of many) to accomplish a goal.

8.1.2 Studying the Good and the Bad

Enlightenment also means knowing the full range of computing- and data-related applications. In school, computing education efforts (understandably) tend to steer young people towards the kinds of applications that are likely to be fun and inspiring to students; and I absolutely believe that supporting creativity and playfulness among young people is incredibly important and worthwhile. However, an overemphasis on the more benign or benevolent applications of computing can inadvertently obscure some of the limitations of these technologies, as well as the very real risks and vulnerabilities that these technologies also produce.

When young people leave the classroom, they're likely to be on SnapChat, YouTube, Instagram, TikTok, or whatever might be around the corner – engaging in all sorts of ways with the platforms around them. How does their computing education help them to reflect on their own participation on these platforms? Do they understand how these systems work, or what is done with their data? Do they understand that data science can be used to both discover how to treat new diseases; *and* as a way to direct fake news to vulnerable people? Should students also learn about how technologies are used for deep fakes, tracking and surveilling groups of people, building weapons, or designing addictive gambling apps? I argue that as the computing education community continues to debate what ought to be "common knowledge" within computing, a useful question to ask might be: how is the goal of producing enlightened citizens advanced by the concepts, practices, and perspectives (Brennan & Resnick, 2012) that we are trying to support?

8.1.3 Understanding Computing- and Data-Related Practices and Ideas

Of course becoming enlightened about data and computing also requires that young people also have the opportunity to learn about the technical. In my view, this is a question of how much and what forms of technical knowledge are necessary within K-12. If reasoning about sociotechnical systems is the goal, then it is important to think about other ways of prioritizing what is important to learn about. If it is not necessary to understand the fine-grained details of CS1, then what other ways might the computing and education communities want to sequence and scaffold learning?

I argue that domain-specific languages (e.g. SQL, HTML) can absolutely be a generative starting point for novices. Not only would developing facility with DSLs enable learners to examine real-world applications and systems, but it may also foster alignment – so that students can better connect their computer science learning activities to the broader systems that surround them. While this approach might not allow learners to reason about how memory is allocated or the most efficient way to sort a list, it does open up new ways to participate in and examine real-world languages and contexts.

8.2 Design Implications

8.2.1 An Anti-Essentialist Agenda

In the pursuit of making computing meaningful, it is important to guard against essentialism – the idea that certain "types" of students find certain types of contexts meaningful, or that "youth" in general enjoy doing any one activity (Sims, 2017). Some arguments within computing education suggest that females and racial minorities are more likely to identify with computing if they perceive it to have social impact or that it ties to traditional cultural practices. While there may indeed be cultural- or gender-specific regularity in the kinds of activities that learners find meaningful (Gutiérrez & Rogoff, 2003), the process of finding meaning looks different for us all.

For instance, within Code 510, Kira and her team – a team of young people who were female, gender-queer, and/or racial minorities – preferred making a zombie game to a "social justice app" or doing an eTextiles project (Chapter 4). Making too many assumptions about what contexts are meaningful to young people can also lead to learning environment designs that are contrived (T. Philip & Garcia, 2013), overly prescriptive, or even paternalistic (T. Philip et al., 2013). By conceptualizing inclusivity as offering multiple opportunities to find meaning within a practice (Nasir & Hand, 2008), we can expose students to many different ideas, applications, and world views (Tippett, 2019), and let students decide for themselves what resonates.

8.2.2 Elevating the Role of "Vocational" Practices

An emphasis on real-world contexts and DSLs also has implications for what is generally considered "foundational" within computing. Instead of emphasizing "computational thinking" (e.g. algorithmic notions of flow of control, structured problem decomposition, listed in 1.1.1), which I argue takes a very computer science-oriented perspective on the common-knowledge computing question, the emphasis would have to shift to supporting skills and practices which are generally considered "vocational" (e.g. declarative programming, dealing with the syntax rules of formal text-based languages, engaging in data collection).

This may be a good trade-off to make in an introductory computing / data course. For people who will eventually go onto become computer scientists (arguably a minority of students), they will be getting an early introduction to the social implications and applications of data. For everyone else: this approach will help them to make sense of how information circulates, the monetary incentives that impact these infrastructures, and the magnitude and scale of personally identifiable information that is available on the Internet.

Learning about particular DSLs can help learners deconstruct familiar artifacts and systems, which may demystify (at least to some extent) the apps and platforms in which they are thoroughly enmeshed. It may also encourage learners to think through some of the benefits, limitations, and risks of various computer-mediated systems. That said, I do not think that this precludes engaging in creative production, design, and all of the other practices that have long been shown to engage learners (Papert, 1980). In fact one of the lessons that I came away with in doing this research is just how important play and imagination are to engagement – at any age. Moreover, building one's own app is a wonderful way to learn, and to think concretely about the mechanisms that might underlie a technology. As I showed in chapter 7, creative production allowed students to tap into existing hobbies to share unique aspects of themselves, envision how they might someday use various computing and data technologies towards their own ends.

8.3 Epilogue

During my time in graduate school, I met hundreds of young people through various research projects, and always wondered what came next for them. After the educational intervention was over and their lives went on, what happened? For most of the students with whom I worked, I don't have an answer to this question, but I have kept up with a few students featured in Chapters 4-6. Anna (p. 116), whose interests had consistently involved social justice related pursuits, went on to major in the humanities. Her brother, Gabriel (p. 99), who had always gravitated to aesthetics and design, majored in interior design with a full scholarship to a prestigious art school. Both are about to graduate with a bachelor's degree. Erick (p. 114), who had participated in several different STEM programs during his high school career, began his undergraduate education as a computer science major. However, after having a few bad experiences with Calculus, he decided to switch to industrial engineering. Ling (p. 113) is on track to graduate with a computer science degree, and did well in her college math classes. When we last spoke, she was applying to summer internships at Bay Area tech firms. When I last spoke to Jocelyn (p. 101), she was was interning at Intel, after completing her second year of community college. She was also majoring in computer science.

These are all great outcomes. That said, the students named above were also unique in many ways. For one, each of them intentionally sought out afterschool STEM enrichment activities and regularly came to *Code 510* every Friday evening, after school, to work on coding projects. This meant they had the time (i.e. they didn't have other obligations that precluded their participation), and were aware of and invested in seeking out extracurricular academic opportunities.

When I co-taught 10th grade Sociology (4.2), I encountered a much more diverse group of students in terms of interests, goals, and life experiences. From students' journal entries and class discussions, I learned about some of the many constraints they regularly navigated: food and housing insecurity, uncertainty regarding their family's legal immigration status, the loss of a parent or loved one, and/or severe bullying. Students also shared experiences that were very much more racialized – being followed around the mall by security guards, seeing their parents get questioned when returning a jacket at a department store, getting harsher disciplinary treatment at school, having their academic struggles misread as them "not caring" or "slacking off."

These impacts were mostly hidden to me in the context of classroom life. However sometimes these stressors would visibly impact students' ability / willingness to engage with the class itself (and school more broadly). Some students would regularly come into the classroom upset, put their heads down on their desks, and refuse to speak for the entire class period. Others students would disappear for weeks at a time - one never came back, and I never knew why. One student was suspended for two weeks for disciplinary reasons, while another joined the class mid-semester after having been expelled from another school. On the other hand, a handful of students never missed a class. There was regularity, but also significant variability in students stories, experiences, and engagements with the class.

These stressors, coupled with students' assessment of the value of school (Fine, 1986; Willis, 1977), also had an impact on students' academic trajectories. Of the 15 students I taught, only seven graduated two years later. Of the other eight, two transferred to and graduated from different high schools, two I could not track down, and four left school altogether (i.e. dropped out / were pushed out). Of these students who left school altogether, two of them were among the highest achieving students in the Sociology class. Contrary to my assumption that leaving school was the result of not having the academic background or knowledge to achieve academically, the story was clearly much more complicated. Obviously, an applied computing course was not going to address any of these issues. All it could do was help students learn a little bit about computing and data.

Community College

When Oba (my mentee, p. 30) began attending community college after earning his high school diploma at the local adult school, I became curious about the computing education program there,[41] and applied to become an adjunct instructor. At the college, I taught a web programming course in what was marketed as California's first public coding "bootcamp." My colleague and I also ran a weekly resume and jobs workshop to help students think about how to break into the industry.

During this time, I had the opportunity to meet a wonderful and diverse group of students – people who were retired, people who were autistic, people pursuing their associate's degree, people with masters/professional degrees, people with GEDs, people who had just moved to the country, etc.

Some of the students were already excellent programmers, loved making apps and websites, and absolutely knew that they wanted to become developers. Lily and Jason fell into this category. Lily, a Thai female in her late thirties who had been in the US for a decade, was motivated, talented, personable, and responsible, and yet she struggled to get an interview for a technology industry job. Phil, an African-American man in his early 20s who earned the highest grade in several of my classes, was another stand out, but also struggled to break into the field. Neither Lily or Jason knew anyone in the technology industry.

[41] Which, as I mentioned in chapter 2, was how I also learned to program.

On the other hand, Melanie and Lynda – white women in their 30s and 40s, with masters degrees from prestigious universities – both had friends working at Google and Facebook. They were in community college only to make a career pivot. While their respective journeys of finding a tech job were still long and stressful, they did eventually find a position as UI/UX researchers at different firms. They were good students *and* they also had other kinds of capital that helped them to break into the industry.

In seeing these experiences, I began to take stock of my own journey, and the taken-for-granted preconditions that had enabled me to tell my "anyone can get a tech job" (2.3.1) story after my own vocational school experience. Years ago, when I was lost and struggling to find my way, I had a lot of help – a flexible part-time job that allowed me to attend classes; a room at my mom's place, rent-free; no family to provide for; and a Yale degree. I was also white and American born. And yet it still felt hard.

Meanwhile, many of the students that I was teaching had young children, were paying their own rent, worked a patchwork of gig jobs (catering, ubering), were learning a second language, and lived in one of the most expensive places in the country.[42] I could only imagine what it was like for them. What had enabled me to land my first programming job so many years ago was not so much my coding skills, as it was an entire infrastructure that had allowed me to take full advantage of an opportunity.

I serendipitously heard from Lily this week over email (unprompted), as I was writing this conclusion. She told me that one of her teachers had nominated her for an apprenticeship program, which she had just completed. She had loved her apprenticeship, which had led to her placement at a local firm as a full-time programmer. After watching Lily struggle for three long years, a non-profit organization had come along and provided her with the requisite infrastructure for her to begin the next leg of her journey – providing her with a stipend, child care if needed, a laptop, an apprenticeship placement, and ultimately an industry placement.

While models like this don't solve other tech-induced ills, they do offer a way to help people to manifest their skills and abilities, giving them access to opportunities and infrastructures that many of us take for granted. I hope this program grows and thrives, because it gets at the very heart of how real change happens – acknowledging and then *providing* the requisite financial support, training, and much-needed social connections. If the technology industry were to offer reparations for all of the jobs that have been automated away, neighborhoods gentrified, taxes not paid, women/racial minorities harassed or overlooked, people surveilled, and personal data sold – supporting programs like this could be one way to do it.

---

[42] One student, after losing his housing, was staying with family members over two hours away, and was commuting to class until he couldn't anymore and dropped out.

REFERENCES

Ames, M. G. (2019). *The charisma machine: The life, death, and legacy of One Laptop per Child*. Cambridge, MA: MIT Press.

Anderson, N., Lankshear, C., Timms, C., & Courtney, L. (2008). "Because it's boring, irrelevant and I don't like computers": Why high school girls avoid professionally-oriented ICT subjects. *Computers and Education*, *50*(4), 1304–1318. https://doi.org/10.1016/j.compedu.2006.12.003

Ben-zvi, D., & Arcavi, A. (2001). Junior High School Students' Construction of Global Views of Data and Data Representations. *Educational Studies in Mathematics*, *45*, 35–65. https://doi.org/10.1023/A:1013809201228

Berman, F., Rutenbar, R., Hailpern, B., Christensen, H., Davidson, S., Estrin, D., … Szalay, A. S. (2018). Realizing the potential of data science. *Communications of the ACM*, *61*(4), 67–72. https://doi.org/10.1145/3188721

Boaler, J. (2008). Promoting 'relational equity' and high mathematics achievement through an innovative mixed-ability approach. *British Educational Research Journal*, *34*(2), 167–194. https://doi.org/10.1080/01411920701532145

boyd, danah. (2007). Why youth (heart) Social network sites: the role of networked publics in teenage social life. *MacArthur Foundation Series on Digital Learning – Youth, Identity, and Digital Media*, *7641*(41), 1–26. https://doi.org/10.1162/dmal.9780262524834.119

boyd, danah, & Crawford, K. (2012). Critical questions for big data: Provocations for a cultural, technological, and scholarly phenomenon. *Information, Communication & Society*, *15*(5), 662–679. https://doi.org/10.1080/1369118X.2012.678878

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. *Proceedings of the 2012 Annual Meeting of …*, 1–25. Retrieved from http://web.media.mit.edu/~kbrennan/files/Brennan_Resnick_AERA2012_CT.pdf

Burke, J., Estrin, D., Hansen, M., Parker, N., Ramanathan, N., Reddy, S., & Srivastava, M. B. (2006). Participatory sensing. *Presented at World Sensor Web Workshop, ACM SenSys '06, Boulder, CO*. Retrieved from https://escholarship.org/uc/item/19h777qd

Cain, C. C. (2012). Underrepresented groups in gender and STEM: the case of black males in CISE. In *50th annual conference on Computers and People Research* (pp. 97–102). ACM.

Calabrese Barton, A., & Tan, E. (2019). Designing for rightful presence in STEM: The role of making present practices. *Journal of the Learning Sciences*, 1–43.

https://doi.org/10.1080/10508406.2019.1591411

Cammarota, J., & Fine, M. (2010). Youth Participatory Action Research. In J. Cammarota & M. Fine (Eds.), *Revolutionizing Education: Youth Participatory Action Research in Motion* (pp. 1–22). Routledge. Retrieved from https://books.google.com/books?id=9_aTAgAAQBAJ&dq=cammarota+%26+fine&source=gbs_navlinks_s

Chambers, R. (2006). Participatory mapping and Geographic Information Systems: Whose map? Who is empowered and who disempowered? Who gains and who loses? *The Electronic Journal of Information Systems in Developing Countries*, *25*(1), 1–11. https://doi.org/10.1002/j.1681-4835.2006.tb00163.x

Chen, E., Chen, S., Tague, P., & Kotcher, R. (2014). OAuth Demystified for Mobile Application Developers. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)* (pp. 892–903). https://doi.org/10.1145/2660267.2660323

Cheryan, S., Master, A., & Meltzoff, A. N. (2015). Cultural stereotypes as gatekeepers: increasing girlsâ€™ interest in computer science and engineering by diversifying stereotypes. *Frontiers in Psychology*, *6*(February), 1–8. https://doi.org/10.3389/fpsyg.2015.00049

Cobb, P., Confrey, J., diSessa, A., Lehrer, R., & Schauble, L. (2003). Design Experiments in Educational Research. *Educational Researcher*, *32*(1), 9–13. https://doi.org/10.3102/0013189X032001009

Code.org. (2013). What Most Schools Don't Teach. Retrieved December 22, 2019, from https://www.youtube.com/watch?v=nKIu9yen5nc

Cole, M. (1996). Putting culture in the middle. In *Cultural psychology: a once and future discipline* (pp. 116–145).

Cole, M., & Engeström, Y. (2007). Cultural-Historical Approaches to Designing for Development. In *The Cambridge Handbook of Sociocultural Psychology* (pp. 484–507). https://doi.org/10.1017/CBO9780511611162.026

Conway, D. (2010). The Data Science Venn Diagram. Retrieved January 1, 2015, from http://drewconway.com/zia/2013/3/26/the-data-science-venn-diagram

Cooper, S., Grover, S., Guzdial, M., & Simon, B. (2014). A future for computing education research. *Communications of the ACM*, *57*, 34–36. https://doi.org/10.1145/2668899

Dasgupta, S., Hale, W., Monroy-Hernández, A., & Hill, B. M. (2016). Remixing as a Pathway to Computational Thinking. *Computer Supported Collaborative Work*, 1438–1449.

Delpit, L. (2006). *Other people's children: Cultural conflict in the classroom*. New York: The New

Press.

Dewey, J. (1938). *Experience and education*. Kappa Delta Pi. Retrieved from https://books.google.com/books?hl=en&lr=&id=JhjPK4FKpCcC&oi=fnd&pg=PA17&dq=experience+and+education&ots=D9BD0tDCCe&sig=tAxysotnBqtPHECbj8fBU4CRqio

Dimond, J. P., Dye, M., LaRose, D., & Bruckman, A. S. (2013). Hollaback!: The role of collective storytelling online in a social movement organization. *Proceedings of the ACM Conference on Computer Supported Cooperative Work, CSCW*, 477–489. https://doi.org/10.1145/2441776.2441831

diSessa, A. A. (2018). Computational Literacy and "The Big Picture" Concerning Computers in Mathematics Education. *Mathematical Thinking and Learning*, *20*(1), 3–31. https://doi.org/10.1080/10986065.2018.1403544

Dorn, B. (2010). A Case-Based Approach for Supporting the Informal Computing Education of End-User Programmers, (December).

Elwood, S. (2006). Beyond cooptation or resistance: Urban spatial politics, community organizations, and GIS-based spatial marratives. *Annals of the Association of American Geographers*, *96*(2), 323–341. https://doi.org/10.1111/j.1467-8306.2006.00480.x

Elwood, S., & Mitchell, K. (2013). Another politics is possible: Neogeographies, visual spatial tactics, and political formation. *Cartographica: The International Journal for Geographic Information and Geovisualization*, *48*(4), 275–292. https://doi.org/10.3138/carto.48.4.1729

Engeström, Y., & Sannino, A. (2010). Studies of expansive learning: Foundations, findings and future challenges. *Educational Research Review*, *5*(1), 1–24. https://doi.org/10.1016/j.edurev.2009.12.002

Enyedy, N., & Mukhopadhyay, S. (2007). They don't show nothing I didn't know: Emergent tensions between culturally relevant pedagogy and mathematics pedagogy. *Journal of the Learning Sciences*, *16*(2), 139–174. https://doi.org/10.1080/10508400701193671

Erickson, F. (2006). Studying Side by Side: Collaborative Action Ethnography in Educational Research. In G. Spindler, L. Hammond, & L. A. Hammond (Eds.), *Innovations in Educational Ethnography: Theory, Methods, and Results* (pp. 235–257). Psychology Press. https://doi.org/10.4324/9780203837740

Eubanks, V. (2018). *Automating inequality: How high-tech tools profile, police, and punish the poor.* New York: St. Martin's Press.

Fine, M. (1986). Why urban adolescents drop into and out of public high school. *Teachers College Record*, *87*(3), 393–409.

Fiore-Gartland, B., & Neff, G. (2015). Communication, mediation, and the expectations of data: Data valences across health and wellness communities. *International Journal of Communication*, *9*(1), 1466–1484.

Freire, P. (1996). *Pedagogy of the oppressed*. New York: Continuum. (Original work published 1970).

Freire, P. (1998). *Pedagogy of freedom: Ethics, democracy, and civic courage*. Lanham, Maryland: Rowman & Littlefield Publishers.

Gillespie, T. (2010). The politics of 'platforms.' *New Media & Society*, *12*(3), 347–364. https://doi.org/10.1177/1461444809342738

Ginsburg, H. P., & Opper, S. (1988). *Piaget's theory of intellectual development*. Prentice-Hall, Inc.

González, N., Andrade, R., Civil, M., & Moll, L. (2001). Bridging Funds of Distributed Knowledge: Creating Zones of Practices in Mathematics. *Journal of Education for Students Placed at Risk (JESPAR)*, *6*(1–2), 115–132. https://doi.org/10.1207/S15327671ESPR0601-2_7

Goode, J., & Margolis, J. (2011). Exploring Computer Science. *ACM Transactions on Computing Education*, *11*(2), 1–16. https://doi.org/10.1145/1993069.1993076

Google Inc., & Gallup Inc. (2016). Trends in the State of Computer Science in U.S. K-12 Schools. Retrieved December 30, 2019, from http://goo.gl/j291E0

Gray, Jeff, Abelson, H., Wolber, D., & Friend, M. (2012). Teaching CS principles with app inventor. *Proceedings of the 50th Annual Southeast Regional Conference on - ACM-SE '12*, 405. https://doi.org/10.1145/2184512.2184628

Gray, Jonathan, Gerlitz, C., & Bounegru, L. (2018). Data infrastructure literacy, (December), 1–13. https://doi.org/10.1177/2053951718786316

Greeno, J. G. (1997). On Claims That Answer the Wrong Questions. *Educational Researcher*, *26*(1), 5–17. https://doi.org/10.3102/0013189X026001005

Grover, S., & Pea, R. (2013). Computational Thinking in K−12 : A Review of the State of the Field. *Educational Researcher*, *42*(1), 38–43. https://doi.org/10.3102/0013189X12463051

Gutierrez, K., Cortes, K., Cortez, A., DiGiacomo, D., Higgs, J., Johnson, P., … Vakil, S. (2017). Replacing Representation With Imagination: Finding Ingenuity in Everyday Practices. *Review of Research in Education*, *41*(1), 30–60. https://doi.org/10.3102/0091732X16687523

Gutiérrez, K. D. (2008). Developing a sociocritical literacy in the third space. *Reading Research*

Quarterly, *43*(2), 148–164. Retrieved from http://onlinelibrary.wiley.com/doi/10.1598/RRQ.43.2.3/abstract

Gutiérrez, K. D., & Jurow, A. S. (2016). Social design experiments: Toward equity by design. *Journal of the Learning Sciences*, *25*(4), 565–598. https://doi.org/10.1080/10508406.2016.1204548

Gutiérrez, K. D., & Rogoff, B. (2003). Cultural Ways of Learning : Individual Traits or Repertoires of Practice. *Educational Researcher*, *32*(5), 19–25. Retrieved from http://edr.sagepub.com/content/32/5/19.short

Gutiérrez, K. D., & Vossoughi, S. (2010). Lifting Off the Ground to Return Anew: Mediated Praxis, Transformative Learning, and Social Design Experiments. *Journal of Teacher Education*, *61*(1–2), 100–117. https://doi.org/10.1177/0022487109347877

Gutierrez, K., Rymes, B., & Larson, J. (1995). Script, counterscript, and underlife in the classroom: James Brown versus Brown v. Board of Education. *Harvard Educational Review*, *65*(3), 445–472. https://doi.org/0017-8055/95/0800-455

Gutstein, E. (2003). Teaching and Learning Mathematics for Social Justice in an Urban, Latino School. *Journal for Research in Mathematics Education*, *34*(1), 37. https://doi.org/10.2307/30034699

Guzdial. (2018, November). Direct instruction is better than discovery, but what should we be directly instructing? *Communications of the ACM*. Retrieved from http://bit.ly/2ARuUTA

Guzdial, M. (2011). Why we ought to teach Java: Computing education and social practice. Retrieved February 12, 2019, from https://computinged.wordpress.com/2011/08/23/why-we-ought-to-teach-java-computing-education-and-social-practice/

Guzdial, M. (2016). *Learner-Centered Design of Computing Education: Research of Computing for Everyone*. Morgan & Claypool Publishers.

Irwin, A. (1995). *Citizen science: A study of people, expertise and sustainable development*. New York: Routledge. https://doi.org/10.5334/opt.101109

Isaac, M. (2019, October 28). Dissent Erupts at Facebook Over Hands-Off Stance on Political Ads. *New York Times*. Retrieved from https://www.nytimes.com/2019/10/28/technology/facebook-mark-zuckerberg-political-ads.html

Ito, M., Baumer, S., Bittanti, M., Boyd, D., Cody, R., Herr-Stephenson, B., … Tripp, L. (2010). *Hanging Out, Messing Around and Geeking Around. MIT Press.* https://doi.org/10.1111/j.1467-8535.2010.01154_5.x

Kafai, Y. B., & Burke, Q. (2014). *Connected Code: Why Children Need to Learn Programming*. Cambridge, MA: MIT Press.

King, J., Lampinen, A., & Smolen, A. (2011). Privacy: Is there an app for that? In *Proceedings of the Seventh Symposium on Usable Privacy and Security (SOUPS)* (pp. 1–20). Pittsburg, PA.

Ladson-Billings, G. (1995). Toward a Theory of Culturally Relevant Pedagogy. *American Educational Research Journal*, *32*(3), 465–491. https://doi.org/10.3102/00028312032003465

Langer-Osuna, J. M. (2011). How Brianna Became Bossy and Kofi Came Out Smart: Understanding the Trajectories of Identity and Engagement for Two Group Leaders in a Project-Based Mathematics Classroom. *Canadian Journal of Science, Mathematics and Technology Education*, *11*(3), 207–225. https://doi.org/10.1080/14926156.2011.595881

Lanouette, K., Van Wart, S., & Parikh, T. S. (2016). Supporting elementary students' learning about ecological systems through data modeling and interactive mapping in local contexts. In C. K. Looi, J. Polman, U. Cress, & P. Reimann (Eds.), *Proceedings of the International Conference of the Learning Sciences (ICLS)* (Vol. 1, pp. 570–577). Singapore.

Lave, J. (1988). *Cognition in Practice: Mind, mathematics, and culture in everyday life*. Cambridge University Press.

Lave, J., & Wenger, E. (1991). *Situated learning: Legitimate peripheral participation*. Cambridge University Press.

Lee, C. H. (2012). Electronic Thesis and Dissertations Los Angeles Re-mastering the Master ' s Tools : Recognizing and affirming the life experiences and cultural practices of urban youth in critical computational literacy through a video game project A thesis submitted in .

Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Ericjson, J., … Werner, L. (2011). Computational thinking for youth in practice. *ACM Inroads*, *2*(1), 32. https://doi.org/10.1145/1929887.1929902

Lee, V., & Drake, J. (2013). Quantified recess: design of an activity for elementary students involving analyses of their own movement data, (IDC), 273–276. https://doi.org/10.1145/2485760.2485822

Lehrer, R. (2007). Introducing Students to Data Representation and Statistics. *Proceedings of the 30th Annual Conference of the Mathematics Educaiton Research Group of Australia*, *1*, 22–41. Retrieved from http://www.merga.net.au/documents/keynote32007.pdf

Lessig, L. (2000). Code is Law. *Harvard Magazine*. Retrieved from https://www.harvardmagazine.com/2000/01/code-is-law-html

Lofland, J., Snow, D. A., Anderson, L., & Lofland, L. H. (2005). *Analyzing social settings: A guide to qualitative observation and anal* (Fourth). Belmont, CA: Wadsworth.

Maloney, J., Peppler, K. a., Kafai, Y. B., Resnick, M., & Rusk, N. (2008). Programming by Choice: Urban Youth Learning Programming with Scratch John. In *SIGCSE* (pp. 367–371). Portland, OR: ACM.

Margolis, J. (2008). *Stuck in the shallow end: Education, race, and computing*. Cambridge, MA: MIT Press.

McNamara, A. (2015). *Bridging the Gap Between Tools for Learning and for Doing Statistics*. Los Angeles. Retrieved from https://escholarship.org/uc/item/1mm9303x

Mishel, L. (2018). *Uber and the labor market: Uber drivers' compensation, wages, and the scale of Uber and the gig economy*. Retrieved from https://www.epi.org/publication/uber-and-the-labor-market-uber-drivers-compensation-wages-and-the-scale-of-uber-and-the-gig-economy/

Morozov, E. (2013). *To save everything click here: The folly of technological solutionism*. New York: PublicAffairs.

Morrell, E. (2002). Toward a critical pedagogy of popular culture: Literacy development among urban youth. Journal of Adolescent &. *Adult Literacy*.

Muibi, H., Dorn, B., & Park, T. H. (2015). Teacher Perspectives on Web Design Instruction, 231–236.

Nardi, B. (1993). *A small matter of programming: perspectives on end user computing*. Cambridge, MA: MIT Press.

Nasir, N. S., & Hand, V. (2008). *From the Court to the Classroom: Opportunities for Engagement, Learning, and Identity in Basketball and Classroom Mathematics*. *Journal of the Learning Sciences* (Vol. 17). https://doi.org/10.1080/10508400801986108

Noble, S. U. (2018). *Algotithms of oppression: How search engines reinforce racism*. New York: NYU Press.

Packer, M. (2001). The Problem of Transfer, and the Sociocultural Critique of Schooling. *The Journal of the Learning Sciences*, *10*(4), 493–514.

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books, Inc. https://doi.org/10.1016/0732-118X(83)90034-X

Patel, N., Chittamuru, D., Jain, A., Dave, P., & Parikh, T. S. (2010). Avaaj Otalo - A field study of an interactive voice forum for small farmers in rural India. *Conference on Human Factors in Computing Systems - Proceedings*, *2*, 733–742. https://doi.org/10.1145/1753326.1753434

Peppler, K. a., & Kafai, Y. B. (2007). From SuperGoo to Scratch: exploring creative digital media production in informal learning. *Learning, Media and Technology*, *32*(2), 149–166. https://doi.org/10.1080/17439880701343337

Perkel, D. (2006). Copy and paste literacy? Literacy practices in the production of a MySpace profile. In K. Drotner, H. S. Jensen, & K. Schroeder (Eds.), *Informal Learning and Digital Media: Constructions, Contexts, Consequences* (pp. 21–23). Newcastle, UK: Cambridge Scholars Press. Retrieved from http://people.ischool.berkeley.edu/~dperkel/writing/perkel_copypasteliteracyDRAFT_August2007.pdf

Philip, T., & Garcia, A. (2013). The Importance of Still Teaching the iGeneration: New Technologies and the Centrality of Pedagogy. *Harvard Educational Review*, *83*(2), 300–320. Retrieved from http://her.hepg.org/index/W221368G1554U158.pdf

Philip, T. M., Schuler-Brown, S., & Way, W. (2013). A framework for learning about big data with mobile technologies for democratic participation: possibilities, limitations, and unanticipated obstacles. *Technology, Knowledge and Learning*, *18*(3), 103–120. https://doi.org/10.1007/s10758-013-9202-4

Philip, T., Way, W., Garcia, A., Schuler-Brown, S., & Navarro, O. (2013). When educators attempt to make "community" a part of classroom learning: The dangers of (mis)appropriating students' communities into schools. *Teaching and Teacher Education*, *34*, 174–183. https://doi.org/10.1016/j.tate.2013.04.011

Rainie, L., & Anderson, R. (2017). Code-Dependent: Pros and Cons of the Algorithm Age. *Pew Research Center*, (February).

Resnick, M., Silverman, B., Kafai, Y., Maloney, J., Monroy-Hernández, A., Rusk, N., … Silver, J. (2009). Scratch. *Communications of the ACM*, *52*(11), 60. https://doi.org/10.1145/1592761.1592779

Richardson, L., & St. Pierre, E. A. (2005). Writing: A method of inquiry. In N. K. Denzin & Y. S. Lincoln (Eds.), *The Sage handbook of qualitative research* (3rd ed., pp. 959–978). Thousand Oaks, CA: Sage.

Rorrer, A. S., Allen, J., & Zuo, H. (2018). A national study of undergraduate research experiences in computing: Implications for culturally relevant pedagogy. *SIGCSE 2018 - Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, *2018-Janua*, 604–609. https://doi.org/10.1145/3159450.3159510

Rosenberg, M., Confessore, N., & Cadwalladr, C. (2018, March 17). How Trump consultants exploited the Facebook data of millions. *New York Times*. Retrieved from https://www.nytimes.com/2018/03/17/us/politics/cambridge-analytica-trump-campaign.html

Rubel, L. H., Hall-Wieckert, M., & Lim, V. Y. (2017). Making space for place: Mapping tools and practices to teach for spatial justice. *Journal of the Learning Sciences*, *26*(4), 643–687. https://doi.org/10.1080/10508406.2017.1336440

Rubel, L. H., Lim, V. Y., Hall-Wieckert, M., & Sullivan, M. (2016). Teaching Mathematics for Spatial Justice: An Investigation of the Lottery. *Cognition and Instruction*, *34*(1), 1–26. https://doi.org/10.1080/07370008.2015.1118691

Ryoo, J. J., Margolis, J., Lee, C. H., Sandoval, C. D. M., & Goode, J. (2013). Democratizing computer science knowledge: transforming the face of computer science through public high school education. *Learning, Media and Technology*, *38*(2), 161–181. https://doi.org/10.1080/17439884.2013.756514

Saldaña, J. (2016). *The coding manual for qualitative researchers* (Third). Thousand Oaks, CA: Sage Publications.

Schneider, J. (2018). The Case for Computer Science as a High School Foreign Language. Retrieved December 30, 2019, from https://www.compuscholar.com/schools/blog/the-case-for-computer-science-as-a-high-school-foreign-language/

Scott, A., Martin, A., McAlear, F., & Madkins, T. C. (2016). Broadening participation in computer science: Existing out-of-school initiatives and a case study. *ACM Inroads*, *7*(4), 84–90. https://doi.org/10.1145/2994153

Scott, J. (1998). *Seeing like a state: How certain schemes to improve the human condition have failed*. New Haven: Yale University Press.

Searle, K. a., Fields, D. a., Lui, D. a., & Kafai, Y. B. (2014). Diversifying high school students' views about computing with electronic textiles. *Proceedings of the Tenth Annual Conference on International Computing Education Research - ICER '14*, 75–82. https://doi.org/10.1145/2632320.2632352

Searle, K. A., & Kafai, Y. B. (2015). Boys' Needlework: Understanding Gendered and Indigenous Perspectives on Computing and Crafting with Electronic Textiles. *Proceedings of the Eleventh Annual International Conference on International Computing Education Research - ICER '15*, 31–39. https://doi.org/10.1145/2787622.2787724

Shirky, C. (2011). The Political of Social Power Media the Public Sphere , and Political Change Technology ,. *Council on Foreign Relations*, *90*(1), 28–41. https://doi.org/10.1080/1369118X.2012.670661

Sims, C. (2017). *Disruptive fixation: School reform and the pitfalls of techno-idealism*. Princeton and Oxford: Princeton University Press.

Singer, N. (2017, June 27). How Silicon Valley Pushed Coding Into American Classrooms.

New York Times. Retrieved from https://www.nytimes.com/2017/06/27/technology/education-partovi-computer-science-coding-apple-microsoft.html

Sklar, E. (2007). *NetLogo, a multi-agent simulation environment*. Cambridge, MA: MIT Press.

Steele, C. M., & Aronson, J. (1995). Stereotype Threat and the Intellectual Test Performance of African Americans. *Journal of Personality and Social Psychology*, *69*(5), 797–811. https://doi.org/10.1037/0022-3514.69.5.797

Sterling, A. (2019, June 15). Millions of jobs have been lost to automation. Economists weigh in on what to do about it. *Forbes*. Retrieved from https://www.forbes.com/sites/amysterling/2019/06/15/automated-future/

Taylor, K. H., & Hall, R. (2013). Counter-mapping the neighborhood on bicycles: Mobilizing youth to reimagine the city. *Technology, Knowledge and Learning*, *18*, 65–93. https://doi.org/10.1007/s10758-013-9201-5

Tew, A. E., & Guzdial, M. (2010). Developing a Validated Assessment of Fundamental CS1 Concepts. In *SIGCSE '10: Proceedings of the 41st SIGCSE Technical Symposium on Computer Science Education* (pp. 97–101).

Tippett, K. (2019). Ta-Nehisi Coates — Imagining a New America. National Public Radio and WBEZ Chicago. Retrieved from https://www.wbez.org/shows/on-being/tanehisi-coates-imagining-a-new-america/32e32ddf-9509-4786-a0bb-22e0ad6d409d

Tissenbaum, M., Sheldon, J., & Abelson, H. (2019). Viewpoint From Computational Thinking to Computational Action. *Communications of the ACM*, *63*(3), 34–36. https://doi.org/10.1145/3265747

Turner, F. (2006). From Counterculture to Cyberculture: Stewart Brand, the Whole Earth Network, and the Rise of Digital Utopianism. Chicago, Illinois: University of Chicago Press. https://doi.org/10.1177/026732310702200306O2

Urban Strategies Council. (2012). Who owns your neighborhood? The role of investors in post-foreclosure Oakland. Retrieved January 2, 2020, from https://urbanstrategies.org/download/who-owns-your-neighborhood-the-role-of-investors-in-post-foreclosure-oakland/

Vakil, S. (2016). *Learning, Identity, and Power: Tensions and Possibilities in Equity-Oriented Computer Science Education*. UC Berkeley.

Van Deursen, A., & Klint, P. (2002). Domain-Specific Language Design Requires Feature Descriptions. *Journal of Computing and Information Technology*, *10*(1), 1. https://doi.org/10.2498/cit.2002.01.01

Van Wart, S., & Parikh, T. S. (2013). Increasing Youth and Community Agency in GIS. *Paper Presented at the Geographic Human-Computer Interaction (GeoHCI) Workshop at SIGCHI*, 1–3. Retrieved from https://www-users.cs.umn.edu/~bhecht/geohci2013ps/GeoHCI2013_Proceedings.pdf

Van Wart, S., Tsai, K. J., & Parikh, T. (2010). Local Ground: A Paper-Based Toolkit for Documenting Local Geo-spatial Knowledge. *Proceedings of the First ACM Symposium on Computing for Development*, 1. https://doi.org/10.1145/1926180.1926194

Van Wart, S., Vakil, S., & Parikh, T. S. (2014). Apps for Social Justice : Motivating Computer Science Learning with Design and Real-World Problem Solving. *Proceedings of the 2014 Conference on Innovation & Technology in Computer Science Education*, 123–128. https://doi.org/10.1145/2591708.2591751

Visintainer, T. (2017). "Scientists do what we do": Empowering Youth of Color as Learners and Doers of Community-Based Scientific Research. In D. Stroupe (Ed.), *Reframing Science Teaching and Learning* (pp. 135–152). New York and London: Routledge.

Vygotsky, L. S. (1967). Play and Its Role in the Mental Development of the Child. *Soviet Psychology*, *5*(3), 6–18. https://doi.org/10.2753/RPO1061-040505036

Vygotsky, L. S. (1978). *Mind in society: The development of higher mental process*. Cambridge, MA: Harvard University Press. https://doi.org/10.2307/j.ctvjf9vz4

Vygotsky, Lev S. (1986). *Thought and Language, Revised Edition*. Cambridge, MA: MIT Press.

Weintrop, D., & Wilensky, U. (2015a). To block or not to block, that is the question: Students' perceptions of blocks-based programming. In *IDC '15: Proceedings of the 14th International Conference on Interaction Design and Children* (pp. 199–208). https://doi.org/https://doi.org/10.1145/2771839.2771860

Weintrop, D., & Wilensky, U. (2015b). Using Commutative Assessments to Compare Conceptual Understanding in Blocks-based and Text-based Programs. *International Computing Education Research Conference (ICER)*, (April 2016), 101–110. https://doi.org/10.1145/2787622.2787721

Wenger-Trayner, E. (2013). The practice of theory: Confessions of a social learning theorist. In V. Farnsworth & Y. Solomon (Eds.), *Reframing educational research: Resisting the "what works" agenda* (p. 11). Abingdon and New York: Routledge. https://doi.org/10.4324/9780203590737

Wenger, E. (1998). *Communities of Practice: Learning, meaning, and identity*. Cambridge, UK: Cambridge University Press.

Willis, P. (1977). *Learning to labour: How working class kids get working class jobs*. New York: Columbia University Press.

Wing, J. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33. https://doi.org/10.1145/1118178.1118215

Winner, L. (1980). Do artifacts have politics? *Daedalus*, *109*(1), 121–136. https://doi.org/10.2307/20024652

Yin, R. K. (2014). *Case study research: Design and methods* (Fifth Edit). Thousand Oaks, CA: Sage Publications.

Zuckerberg, M. (2017). Bringing the World Closer Together. Retrieved December 22, 2019, from https://www.facebook.com/notes/mark-zuckerberg/bringing-the-world-closer-together/10154944663901634/

APPENDIX A: SELECTED STUDENT RESPONSES TO PRE-INTERNSHIP QUESTIONNAIRE

What is the cloud?

Clouds are white puffs in the sky
A place to store pictures from your phone and computer.
It's literally a room with servers that hold all information / data that people use in cloud storage.
An intangible place where information / data is stored in. It also transmits data in and out of 'the cloud' / database.
Collection of servers used to store and transfer info wirelessly.
All data that is saved to the internet / servers for any and all to access at any time.

What is a web browser? List some examples.

The web browser is the search injon. Google, Bing, MSN.
A web browser is a tab or page on the web you have opened or is common to look up things. Google, Yahoo, Bing.
It is the beginning few words of a website. www, edu, gov
A program that lets you get to the internet. Google Chrome, Firefox, Internet Explorer

What is data?

Data is a lot of information gather up [sic] and collected in a base.
Data is information?
Data is simply any information.
Data is information that ppl use in many ways and can be in many forms.
Basically binary codes
Data is how much space you have on your computer.

What is HTML?

HTML is a type of coding language used to program.
Programming language for web design
I think is based off the Internet and web pages.
HTML is a scripting language used to create basic web pages.
CODING!
One of the coding languages of web pages.
HTML is a language of code.

What is CSS?

CSS is a coding word language.
CSS is used for programming.
CSS is to style HTML they both go along.
IDK (4 people)
CSS or cascading style sheets is used to add functionality and design to web pages.
A way to change fonts in coding.
Another language of web pages (Cascading Style Sheets)

What is JavaScript?
JavaScript is used for the design aspect of technology I think.
JavaScript adds functionality to basic web pages.
Don't know
JavaScript is a language that works differently from HTML
Programming language used to make websites interactive.
Ways to make popups in HTML